

## **LỜI CAM ĐOAN**

Tôi xin cam đoan luận văn thạc sĩ công nghệ thông tin “**HỆ THỐNG ĐO TỐC ĐỘ PHƯƠNG TIỆN GIAO THÔNG TRÊN QUỐC LỘ BẰNG CAMERA KỸ THUẬT SỐ**” là do tôi nghiên cứu, tổng hợp và thực hiện.

Toàn bộ nội dung luận văn, những điều được trình bày là của chính cá nhân tôi hoặc là được tham khảo, tổng hợp đều được trích xuất với nguồn gốc rõ ràng. Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

TP.HCM, ngày 07 tháng 07 năm 2022

**Học viên thực hiện luận văn**

**LÝ BĂNG**

## LỜI CẢM ƠN

Lời đầu tiên, học viên xin bày tỏ lòng biết ơn sâu sắc đến **PGS.TS Vũ Đức Lung**, người đã trực tiếp định hướng và hướng dẫn tận tình học viên trong suốt quá trình hoàn thành luận văn. Những kinh nghiệm của thầy là tiền đề để giúp học viên mở rộng kiến thức và hoàn thành khóa luận tốt nghiệp.

Em cũng xin dành lời cảm ơn chân thành đến Thầy Cô Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM. Các Thầy Cô đã dạy bảo và luôn tạo điều kiện tốt nhất cho học viên trong suốt quá trình tham gia lớp cao học.

Xin trân trọng cảm ơn ban **Giám Đốc trung tâm công nghệ thông tin - viễn thông Tây Ninh**, đã cho phép tôi sử dụng dữ liệu camera thông minh của VNPT Tây Ninh. Cảm ơn các bạn đồng nghiệp cơ quan, đã tạo điều kiện về thời gian và quan tâm động viên tinh thần trong thời gian học viên đi học và hoàn thành luận văn.

Cuối cùng, xin cảm ơn gia đình và bạn bè đã luôn bên tôi, cổ vũ và động viên tôi trong suốt quá trình hoàn thành luận văn này.

TP.HCM, ngày 07 tháng 07 năm 2022

**Học viên thực hiện luận văn**

**LÝ BĂNG**

## MỤC LỤC

<b>LỜI CAM ĐOAN .....</b>	<b>i</b>
<b>LỜI CẢM ƠN .....</b>	<b>ii</b>
<b>DANH MỤC CÁC THUẬT NGỮ, CÁC CHỮ VIẾT TẮT .....</b>	<b>vi</b>
<b>DANH SÁCH BẢNG .....</b>	<b>vii</b>
<b>DANH SÁCH HÌNH VẼ.....</b>	<b>viii</b>
<b>MỞ ĐẦU .....</b>	<b>1</b>
1. Lý do chọn đề tài.....	1
2. Tổng quan về vấn đề nghiên cứu .....	2
3. Câu hỏi nghiên cứu .....	3
4. Mục đích nghiên cứu.....	3
5. Đối tượng và phạm vi nghiên cứu.....	3
6. Phương pháp nghiên cứu.....	4
7. Những đóng góp của đề tài .....	4
<b>Chương 1: TỔNG QUANG NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC .....</b>	<b>5</b>
1.1. Giới thiệu.....	5
1.2. Các nghiên cứu trong nước .....	6
1.3. Các nghiên cứu ngoài nước.....	7
<b>Chương 2: CƠ SỞ LÝ THUYẾT .....</b>	<b>10</b>
2.1. Các kỹ thuật xử lý ảnh và nhận diện đối tượng. ....	10
2.1.1. Tổng quan về một hệ thống xử lý ảnh.....	10
2.1.2. Một số khái niệm.....	12
2.1.3. Các vấn đề cơ bản trong xử lý ảnh.....	14
2.1.4. Nhận diện và phân loại ảnh.....	15
2.2. Giới thiệu về OpenCV .....	15

2.2.1. Tổng quan về OpenCV.....	15
2.2.2. Cấu trúc OpenCV .....	16
2.2.3. Các ứng dụng OpenCV.....	16
2.2.4. Chức năng OpenCV.....	17
2.3. Các Phương pháp phát hiện chuyển động (Sử dụng các kĩ thuật trừ nền trong camera giám sát).....	17
2.3.1. Phát hiện đối tượng chuyển động là gì ?.....	17
2.3.2. Phát hiện các vùng ảnh nổi .....	18
2.3.3. Xử lý các vùng ảnh nổi .....	19
2.3.4. Các kĩ thuật trừ ảnh .....	19
2.3.5. Một số kĩ thuật trừ nền cơ bản.....	22
2.3.6. Kĩ thuật trừ nền nâng cao .....	24
2.4. Phương pháp theo vết đối tượng .....	26
2.4.1. Theo vết đối tượng .....	26
2.4.2. Quy trình theo vết đối tượng.....	27
2.4.3. Các Phương pháp dựa trên Deep Learning .....	29
2.4.4. Các phương pháp truy vết nhiều vật thể trong video .....	34
<b>Chương 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....</b>	<b>39</b>
3.1. Hệ thống phát hiện chuyển động, theo vết phương tiện giao thông và tính tốc độ phương tiện .....	39
3.2. Phương pháp phát hiện chuyển động, theo vết phương tiện giao thông và tính tốc độ phương tiện .....	41
3.3. Xây dựng chương trình .....	45
3.4. Công cụ thực hiện .....	49
<b>Chương 4: THỬ NGHIỆM VÀ ĐÁNH GIÁ .....</b>	<b>51</b>
4.1. Thu thập và tiền xử lý dữ liệu. ....	51
4.2. Cài đặt và thử nghiệm. ....	51

4.2.1. <i>Thiết lập cấu hình</i> .....	51
4.2.2. <i>Dữ liệu thực nghiệm</i> .....	52
4.2.3. <i>Quá trình và kết quả thực nghiệm.</i> .....	52
4.3. <i>Kết luận chương</i> .....	58
<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b> .....	<b>59</b>
<b>DANH MỤC TÀI LIỆU THAM KHẢO</b> .....	<b>60</b>
<b>PHỤ LỤC</b> .....	<b>63</b>

## DANH MỤC CÁC THUẬT NGỮ, CÁC CHỮ VIẾT TẮT

<u>Viết Tắt</u>	<u>Tiếng Anh</u>
<b>OpenCV</b>	Open Source Computer Vision Library
<b>CCD</b>	Charge Coupled Device
<b>CCIR</b>	Consultive Committee for International Radio
<b>CUDA</b>	Compute Unified Device Architecture
<b>GMM</b>	General Method of Moments
<b>LPR</b>	License Plate Recognition
<b>SaaS</b>	Software-as-a-Service
<b>MOG</b>	Mixture of Gaussians
<b>R-CNN</b>	(Region with CNN feature)
<b>SSD</b>	Single-Shot Multibox Detector
<b>SPP-net</b>	spatial pyramid pooling layer

## DANH SÁCH BẢNG

Bảng 1: So sánh các mô hình dựa trên R-CNN .....	33
Bảng 2: Kết quả so sánh vận tốc từ camera bắn tốc độ thật của CSGT và vận tốc đo được của chương trình .....	56

## DANH SÁCH HÌNH VẼ

Hình 1: Các giai đoạn chính trong xử lý ảnh .....	11
Hình 2: Các quá trình phát hiện chuyển động.....	17
Hình 3: Phát hiện vùng ảnh nổi.....	18
Hình 4: Quy trình theo vết đối tượng.....	27
Hình 5: Mô hình R-CNN.....	30
Hình 6: Mô hình SPP-net .....	31
Hình 7: Mô hình mạng Fast R-CNN.....	31
Hình 8: Mô hình mạng Faster R-CNN.....	32
Hình 9: Mô hình mạng SSD.....	33
Hình 10: Nguyên Lý Hoạt Động của IoUTracker.....	35
Hình 11: Sơ Đồ Phương Pháp DEEPSORT.....	37
Hình 12: Sơ đồ phương pháp CenterTrack[24] .....	38
Hình 13: Mô hình hệ thống phát hiện chuyển động, theo vết phương tiện giao thông và tính tốc độ phương tiện.....	39
Hình 14: Mô hình khung đo .....	43
Hình 15: Tỷ lệ chiều dài thực và chiều dài khung đo [3].....	43
Hình 16: Tính vận tốc trung bình của các giai đoạn [1] .....	44
Hình 17: Lưu đồ hoạt động .....	46
Hình 18: Minh họa xác định đối tượng .....	47
Hình 19 : Lưu đồ thuật toán tính khoảng cách di chuyển thực tế của hai đối tượng trên các frame .....	48
Hình 20: Giao diện chương trình chạy video .....	49
Hình 21: Camera quay quét.....	52



Hình 22: Phát hiện và theo vết đối tượng chuyển động.....	54
Hình 23: Kết quả thử nghiệm đo tốc độ.....	55

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong thời đại ngày nay công nghệ thông tin hầu như đã thâm nhập vào toàn bộ các lĩnh vực đời sống xã hội, cùng với đó sự phát triển về mọi mặt của đời sống xã hội từ văn hóa, giáo dục cho đến công nghệ và đặc biệt sự phát triển vượt bậc của thời đại số hóa, công nghiệp từ đó công nghệ thông tin đã và đang góp phần to lớn cho xã hội con người rất nhiều trong nhiều lĩnh vực như kinh tế, đời sống, giáo dục, dịch vụ,... Từ đó nhiều hệ thống camera để theo dõi giám sát an ninh, tình hình giao thông kết hợp với giám sát đã được lắp đặt ngày càng nhiều. Từ những dữ liệu có sẵn nhiều ứng dụng thu thập khai thác phân tích dữ liệu từ hệ thống giám sát theo dõi này như tính toán dự đoán mật độ lưu thông, đếm lưu lượng xe, truy vết, nhận dạng biển số, đo tốc độ xe...

Cùng với đó việc đưa ứng dụng công nghệ thông tin vào việc quản lý kiểm soát, xử phạt hay điều chỉnh các tuyến đường bố trí lại luồng phương tiện tham gia giao thông góp phần kiềm chế những vụ tai nạn giao thông không đáng có cũng như duy trì trật tự, an ninh, an toàn, nâng cao tự giác ý thức trong tham gia giao thông góp phần nhằm điều chỉnh nhận thức cũng như hành vi của người tham gia giao thông là cần thiết.

Trên thực tế cho thấy, mật độ giao thông và ý thức điều khiển tốc độ di chuyển của nhiều phương tiện của người tham gia giao thông ảnh hưởng rất lớn đến xác suất xảy ra tai nạn làm ảnh hưởng đến mức độ nghiêm trọng của mỗi vụ tai nạn. Trong bối cảnh điều kiện hạ tầng giao thông nước ta còn chưa phát triển, tổ chức lưu thông các phương tiện ở nước ta còn nhiều hạn chế cùng với việc người tham gia điều khiển phương tiện giao thông thiếu ý thức, thường xuyên không làm chủ được tốc độ, môi trường tham gia giao thông xấu, đã dẫn đến tình trạng xảy ra các vụ tai nạn giao thông nghiêm trọng ngày càng gia tăng. Vì vậy cần thiết phải các kế hoạch đề ra cải tiến trình quản lý, phải ứng dụng công nghệ thông tin vào việc xây dựng quản lý và giám

sát giao thông thông minh hỗ trợ các cơ quan ban ngành và đó chính là lý do thực hiện đề tài này.

## **2. Tổng quan về vấn đề nghiên cứu**

Thời đại ngày nay từ những nhu cầu cần thiết về xử lý hình ảnh có rất nhiều nghiên cứu về lĩnh vực thị giác máy tính là một trong những lĩnh vực khoa học máy tính được nhiều nhà nghiên cứu trên thế giới và Việt Nam đã và đang quan tâm đặc biệt về tình trạng giao thông, xử lý vi phạm thông qua xử lý hình ảnh, nhận dạng, truy vết và phân loại phương tiện giao thông.

Nghiên cứu về việc xây dựng một hoặc nhiều hệ thống giám sát, truy vết, phân loại phương tiện giao thông và dự đoán đưa ra cảnh báo về tốc độ trong bối cảnh Việt Nam là điều cấp thiết và có ý nghĩa to lớn áp dụng cả về lý thuyết và thực tiễn.

Nhiều nghiên cứu xây dựng hệ thống tập trung giám sát, truy vết, phân loại các phương tiện tham gia giao thông và dự đoán tốc độ dựa trên video là xét xem tại một thời điểm, trong vùng quan tâm, có những loại phương tiện gì, hướng di chuyển, số lượng tương ứng. Về mặt ứng dụng, hệ thống giám sát thuộc nhóm các ứng dụng liên quan đến giao thông thông minh. Việc xây dựng hệ thống giám sát được tích hợp ứng dụng nhiều trong nhiều lĩnh vực quản lý giao thông, trong nhiều hoạt động an ninh, quốc phòng, kinh tế xã hội, như cảnh báo trộm, cảnh báo cháy, hỗ trợ giám sát bảo vệ các mục tiêu quan trọng, góp phần xây dựng an ninh xã hội cùng với đó ứng dụng công nghệ thông tin hỗ trợ cơ quan ban ngành thu thập các chứng cứ tại những tình huống nhạy cảm.

Các yêu cầu cơ bản của việc xây dựng hệ thống giám sát nhận diện, truy vết, phân loại phương tiện giao thông và cảnh báo tốc độ phương tiện là với dữ liệu đầu vào là video giao thông, yêu cầu đầu ra là loại phương tiện tham gia giao thông trong vùng quan sát, thông tin về tốc độ của phương tiện tham gia giao thông (ô tô, xe máy và các phương tiện thô sơ khác).

### 3. Câu hỏi nghiên cứu

Xây dựng hệ thống đo tốc độ phương tiện giao thông trên địa bàn tỉnh Tây Ninh mang lại lợi ích gì cho các sở ban ngành?

Cần những dữ liệu nào phù hợp, để có thể quản lý và dự báo?

Sử dụng Phương pháp nào để phát hiện, nhận diện đối tượng chuyển động.

Ứng dụng phép toán nào để dự đoán hướng di chuyển và tính toán vận tốc của phương tiện tham gia giao thông.

### 4. Mục đích nghiên cứu

Luận văn tập trung nghiên cứu sâu về các lý thuyết liên quan về xử lý ảnh, nhận dạng phân loại và theo vết đối tượng, phát hiện và truy vết đối tượng qua từng frame của video và cuối cùng là dự đoán tốc độ,.

Nghiên cứu các lý thuyết liên quan bài toán tính tốc độ phương tiện

Xây dựng thử nghiệm ứng dụng đo tốc độ ....

Từ các kết quả nghiên cứu học hỏi về lý thuyết, từ đó xây dựng một phần mềm thử nghiệm thực tế áp dụng cho địa phương.

### 5. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu ở đây chủ yếu là các loại phương tiện trên đường phố, camera kỹ thuật số giám sát giao thông.

Các phương pháp nhận dạng đối tượng, theo vết đối tượng và ước lượng tốc độ đối tượng.

Phạm vi nghiên cứu tập trung của luận văn này chỉ tập trung vào đo tốc độ chuyển động của ô tô, Môi trường đặt camera trong điều kiện môi trường ban ngày, sáng và khá rõ.

## **6. Phương pháp nghiên cứu**

- Tập trung Thu thập nghiên cứu các bài báo, phân tích nhiều loại tài liệu về các nghiên cứu đã thực hiện trên thế giới và Việt nam tập trung các vấn đề giám sát thông minh bằng hình ảnh.
- Phân tích, lựa chọn giải pháp và hiện thực thử nghiệm Đánh giá kết quả và hiệu chỉnh nếu có.
- Thu thập dữ liệu
- Hiện thực thử nghiệm và đánh giá kết quả

## **7. Những đóng góp của đề tài**

Vận dụng các thuật toán dự báo để xây dựng hệ thống đo tốc độ trên quốc lộ và các cửa ngõ quan trọng trên địa bàn tỉnh Tây Ninh.

Xây dựng hệ thống đo tốc độ phương tiện tham gia giao thông

# **Chương 1: TỔNG QUANG NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC**

## **1.1. Giới thiệu**

Hiện nay tại các nước phát triển, việc áp dụng sử dụng nhiều camera bắn tốc độ (speeding camera) khá thông dụng với hàng nghìn camera bắn tốc độ và camera chụp hình sai phạm giao thông sai làn đường, vượt đèn đỏ được bố trí nhiều nơi kín đáo và chằng chịt trên khắp mọi nẻo đường.

Hệ thống xác định phương tiện tham gia giao thông không phải là hệ thống mới, tuy nhiên vẫn còn nhiều thách, đặc biệt đối với tình trạng giao thông nước ta có nhiều thách thức như: Tình trạng giao thông ở Việt Nam phức tạp bởi mật độ lưu thông xe máy tăng nhanh một số thời điểm, nhiều loại xe có kích cỡ quá tải, môi trường và điều kiện thời tiết xấu, góc đặt máy quay đa dạng v.v... Bên cạnh đó, áp dụng để có thể phát hiện những trường hợp chạy vượt quá tốc độ quy định, nhiều tuyến đường đã được trang bị các hệ thống máy bắn tốc độ sử dụng laser. Thiết bị này có chi phí rất cao, phải bảo trì bảo dưỡng và việc trang bị số lượng nhiều trên nhiều tuyến đường là không khả thi, vì vậy ta có thể tận dụng hình ảnh thu được từ các video giám sát giao thông từ đó sử dụng các phương pháp xử lý ảnh để phát hiện phân tích tính toán tốc độ xe đang là giải pháp tối ưu được nhiều nhà nghiên cứu quan tâm. Tóm lại, để tăng cường việc giám sát phát hiện, truy vết, xử lý các sai phạm của phương tiện tham gia giao thông qua các camera giám sát đang được trang bị với số lượng lớn trên các con đường khắp cả nước, hướng tiếp cận áp dụng các kỹ thuật thuật toán xử lý ảnh để phát hiện, truy vết và tính toán tốc độ các phương tiện là cần thiết và khả thi với mức chi phí hợp lý.

## 1.2. Các nghiên cứu trong nước

Ngày nay ở nước ta, các nghiên cứu về vấn đề nhận diện, đo tốc độ xe chuyển động qua camera còn rất nhiều hạn chế do tính phức tạp, mật độ giao thông và ứng dụng vào thực tế. Chủ yếu là tập trung nhiều vào các phương pháp nghiên cứu liên quan về bám đối tượng trong video, dự đoán hướng di chuyển... Hiện nay nhiều bài báo nghiên cứu khoa học có nhiều thành viên tham gia học hỏi nghiên cứu phát triển như: Năm 2015 các tác giả Nguyễn Văn Hùng, Nguyễn Văn Xuất, Lê Mạnh Cường [1] dùng phương pháp một mô hình xác suất có sử dụng hai đặc trưng đó là dựa vào bề mặt và màu sắc để đi tìm đối tượng trong các khung ảnh mới, phương pháp này có theo sát truy vết các đối tượng hiệu quả trong các cảnh khác nhau.

Năm 2016 có nhiều tác giả đưa ra nhiều nghiên cứu điển hình như Lâm Hữu Tuấn, Nguyễn Thị Hồng Nhung, Trần Cao Đệ, Huỳnh Phụng Toàn [2] sử dụng phương pháp xác định tính toán vận tốc chuyển động của các xe từ hình ảnh video với một “khung đo” để nhận biết phát hiện đối tượng chuyển dựa trên phương pháp trừ nền, được ứng dụng phép toán tính trung bình có trọng số để tính toán và đoán hướng di chuyển và vận tốc của đối tượng.

Công trình của Viet-Hoa Do và các cộng sự “A simple camera calibration method for vehicle velocity estimation” [3]. Xác định các tham số camera (camera calibration). Mục tiêu của việc này là nhằm xác định được tọa độ thực tế của một pixel trong ảnh thu được bởi camera. Để thực hiện được điều này, camera cần phải xác định ba thông số sau: độ cao  $h$ , góc nghiêng (tilt angle)  $\theta$  và khoảng cách lấy nét (focus distance)  $f$ .

Tại hội nghị FAIR 2020 tác giả **Vũ Đức Lung** và các cộng sự “khảo sát bài toán nhận diện phương tiện và đo tốc độ phương tiện tham gia giao thông” [4]. Trong bài toán này tác giả đã nêu lên được các phương pháp trong việc nhận diện và đo tốc độ phương tiện giao thông.

### 1.3. Các nghiên cứu ngoài nước

Hiện nay, nhiều nghiên cứu ngoài nước về các kỹ thuật trừ nền, sử dụng camera hỗ trợ trong việc truy vết và camera phân tích xử lý ảnh để đo tốc độ.

Công trình của Jakub Sochor và các cộng sự "Comprehensive Dataset for Automatic Single Camera Visual Speed Measurement" [8]. Nhóm tác giả đề ra 1 tập dataset và đánh giá 1 số phương pháp trên tập dataset này.

Các hướng tiếp cận chính:

- + Dựa trên các line markings: quan sát các line markings. Chỉ áp dụng được khi các line markings này được đặc tả, có hiển thị và nhận diện. Phương pháp này chỉ giới hạn ở đường nội thành. Và 1 số phương pháp yêu cầu đo đạc con đường cũng là 1 bất lợi

- + Dựa trên các di chuyển của phương tiện: không cần các line markings. Tuy nhiên vẫn chỉ được sử dụng ở đường nội thành nhỏ khi mà các hiệu chỉnh (calibration) cũng tốn rất nhiều thời gian khi phải quan sát các phương tiện

- + Tính toán thủ công: độ chính xác cao (trên 1 số cases), tuy nhiên phải dùng các phương tiện giao thông để làm phép đo đạc

- + Dựa trên thống kê các chiều không gian: tính toán trên 2 vanishing point khác nhau"

Công trình của Dominik Zapletal và các cộng sự "Vehicle Re-Identification for Automatic Video Traffic Surveillance" nhận diện phương tiện (cột H) với đề xuất (cột F) sử dụng 3D bounding box, chỉ dùng mặt bên và mặt trước của vật thể (ko dùng hình ảnh từ đỉnh xe), sau đó ghép 2 phần lại thành 1 ảnh đại diện cho vật thể đó và trích xuất đặc trưng màu sắc từ ảnh đó \* Proposed vehicle Re-ID: dùng linear SVM classifier cho kết quả True positive rate khoảng 60% [9]

Công trình của Zheng Tang, và các cộng sự "single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic



features” [10] bài báo này tác giả sử dụng 2 phương pháp đơn camera 3D và đa camera để xác định phương tiện, phân loại phương tiện và ước lượng tốc độ. Bài báo được top 1 cuộc thi AI city 2018

Công trình của Ahmad Arinaldiet al “Detection and classification of vehicles for traffic video analytics” Tác giả đề xuất một hệ thống phân tích giao thông dựa trên thị giác máy tính. Hệ thống có khả năng đếm số phương tiện, phân loại phương tiện, ước lượng tốc độ và theo dõi quá trình sử dụng làn đường. [11]

Năm 2019, Fukai Zhang [12] và các cộng sự đã đề xuất mô hình Single-Shot Multibox Detector (SSD) để phát hiện nhận diện các phương tiện tham gia thông và có khả năng nhận dạng phân loại phát hiện nhiều loại phương tiện khác nhau trong thời gian thực. Nhóm tác giả đã ứng dụng nhiều bộ trích xuất đặc trưng khác nhau trong việc xác định các phương tiện, vị trí và phân loại trong cùng một mạng CNN. Để cải tiến hoàn thiện chất lượng, các đặc trưng này được kết hợp với nhau thông qua các thao tác deconvolution và pooling. Nhóm tác giả cũng tiến hành điều chỉnh các tỷ lệ khác nhau thông qua bounding box để có thể phát hiện các phương tiện có kích thước nhỏ được chính xác hơn. Nhìn chung, hệ thống nhận diện phương tiện giao thông có thể được chia thành 3 phương pháp tiếp cận chính bao gồm: phương pháp dựa trên các đặc trưng chuyển động, phương pháp dựa trên các đặc trưng cấp thấp và phương pháp dựa trên mạng neural.

Tác giả Kenneth S. Palacio-Baus [13] từ Đại học Illinois tại Chicago theo dõi đo tốc độ xe và truy vết ô tô trên cao tốc sử dụng thuật toán trừ nền MOG2 từ thư viện OpenCV và các thư viện công cụ khác cho kết quả nhận diện đối tượng tốt. Tại Thái Lan [14] công ty SmartVision Technology với công nghệ (Smart - LPR) đây là một trong số nhiều sản phẩm thương mại, có khả năng đọc biển số xe tự động, theo vết phương tiện, với tốc độ đo tối đa 180 km/h có độ chính xác 85% trở lên. Công ty Traffiko [15] với nhiều giải pháp vận chuyển thông minh, sáng tạo, cải tiến, cung cấp nhiều ứng dụng vào web để quản lý bãi đỗ xe bằng SaaS thông minh và các ứng dụng dựa trên, thực thi pháp luật liên quan đến giao thông đường phố và ngoài phố thông

qua phân tích thời gian hành trình, phân tích kiểm soát truy cập video dựa trên một nền tảng tích hợp...

Trong quá trình học, chúng tôi nghiên cứu các bài báo của nước ngoài và các bài báo của Việt Nam. Hầu hết các bài báo của nước ngoài đều đề cập hiệu quả việc phát hiện phương tiện bằng một số giải pháp khác nhau, đặc biệt là học sâu. Ngoài ra, đã có các loại giấy của Việt Nam giải quyết vấn đề này bằng cách xử lý đường viền. Vì vậy, chúng tôi nhận thấy rằng việc áp dụng phương pháp học sâu để phát hiện phương tiện cho giao thông Việt Nam là cần thiết.

## **Chương 2: CƠ SỞ LÝ THUYẾT**

Trong chương này luận văn sẽ giới thiệu cơ bản về các khái niệm, đặc trưng và các kỹ thuật từ cơ bản đến nâng cao liên quan đến xử lý hình ảnh và phát hiện đối tượng. Các kỹ thuật hiện tại áp dụng cho hệ thống phát hiện, truy vết đối tượng, đo tốc độ đối tượng và những hạn chế của các kỹ thuật này.

### **2.1. Các kỹ thuật xử lý ảnh và nhận diện đối tượng**

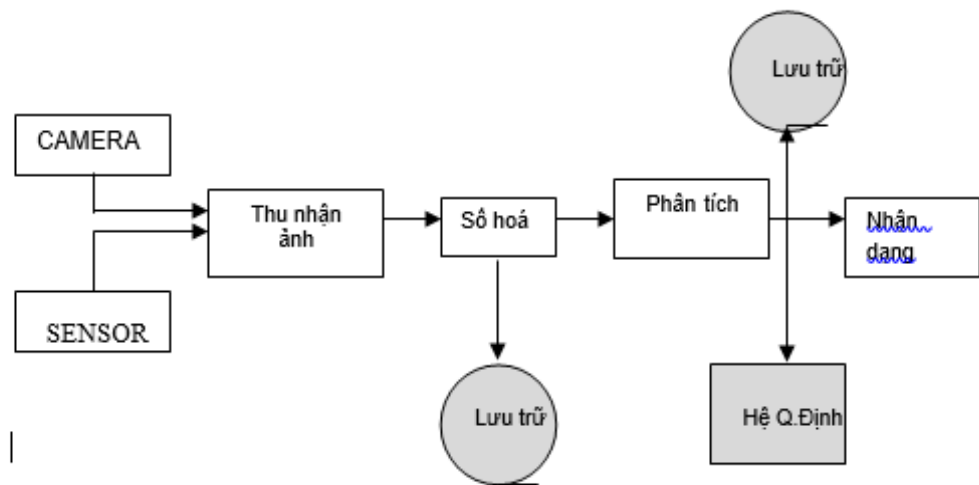
#### ***2.1.1. Tổng quan về một hệ thống xử lý ảnh***

Việc thu thập xử lý phân tích, nhận dạng hình ảnh là đối tượng nghiên cứu của lĩnh vực thị giác máy, là một quá trình biến đổi từ một ảnh hoặc nhiều ảnh ban đầu sang một ảnh mới với các đặc tính, đặc trưng và luôn tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm nhiều quá trình thu thập phân tích dữ liệu đầu vào nhận từ các camera giám sát, từ phân lớp các đối tượng, làm tăng chất lượng, hoặc giảm, phân đoạn và tách cạnh, thông tin hình ảnh của ảnh ta có thể gán nhãn cho vùng hay quá trình biên dịch. Việc xử lý bộ dữ liệu hình ảnh bằng đồ họa, xử lý ảnh số từ các video là một trong nhiều lĩnh vực của tin học ứng dụng. Một cấu trúc dữ liệu và được tạo bởi nhiều chương trình ta xem như là các tập dữ liệu hình ảnh. Xử lý ảnh nhân tạo bằng nhiều tập dữ liệu đồ họa đề cập đến, Các kỹ thuật xử lý ảnh số là tập hợp nhiều phương pháp và kỹ thuật biến đổi, truyền tải hoặc mã hóa các ảnh tự nhiên. Mục đích của việc xử lý ảnh gồm: Biến đổi ảnh làm giảm chất lượng ảnh hoặc tăng chất lượng ảnh. Tự động đón nhận ảnh, nhận dạng ảnh, đánh giá lại các nội dung của ảnh. Từ đó việc nhận biết và đánh giá các nội dung của nhiều tập ảnh là sự phân tích một hay nhiều hình ảnh thành những phần có ý nghĩa để phân biệt để phân biệt nhận biết đối tượng này với một hoặc nhiều đối tượng khác, dựa vào đó ta có thể mô tả lại các cấu trúc của hình ảnh ban đầu. Ta có thể liệt kê nhiều phương pháp nhận dạng, phân tích ảnh cơ bản như phân biệt nhận dạng ảnh của các đối tượng trên ảnh một hoặc nhiều, tách cạnh, phân đoạn hình ảnh,... Các kỹ thuật này được tập trung sử

dụng nhiều nhất trong y học (xử lý tế bào, nhiễm sắc thể), nhận dạng hình chữ trong văn bản.

Trước hết là quá trình thu nhận ảnh. Ảnh có thể thu nhận qua camera. Thông thường ảnh được thu nhận ghi lại qua camera là tín hiệu số hóa (loại CCD - Charge Coupled Device), và cũng có thể là tín hiệu tương tự (loại camera ống kính CCIR)

- **Các quá trình xử lý ảnh**



**Hình 1: Các giai đoạn chính trong xử lý ảnh**

(Nguồn: Giáo trình xử lý ảnh [5])

Ngoài ra Trong các trường hợp ảnh cũng có thể thu nhận từ vệ tinh đặc trưng riêng qua các bộ cảm ứng (sensor), tranh được quét trên scanner, hay ảnh. Trong quá trình số hóa (Digitalizer) để được biến đổi từ tín hiệu tương tự được chuyển sang tín hiệu rời rạc (lấy mẫu) và được số hóa bằng lượng hóa, trước khi được chuyển sang giai đoạn sau tiền xử lý, phân tích hay lưu trữ lại.

Quá trình phân tích ảnh thực chất phân chia ra nhiều công đoạn nhỏ. Trước hết là phải tăng cường ảnh để nâng cao chất lượng ảnh. Trên thực tế có nhiều nguyên nhân khác nhau: có thể do nguồn sáng do chất lượng thiết bị thu nhận ảnh, hay do nhiễu, chất lượng đường truyền không tốt, ảnh có thể bị suy biến. Vì vậy cần phải có những thuật toán, thiết bị, tăng cường và khôi phục lại ảnh để làm nổi bật một số đặc tính chính của ảnh để dễ nhận biết và phân biệt, sẽ làm cho ảnh gần giống nhất với

trạng thái gốc - trạng thái trước khi ảnh bị biến dạng và được xử lý. Và giai đoạn tiền xử lý là phát hiện các đặc tính đặc trưng như biên, phân vùng ảnh, trích chọn các đặc tính, v.v...

Tùy theo các mục đích sử dụng của ứng dụng<sup>7</sup>, trong giai đoạn nhận dạng, phân lớp hay các quyết định khác. Quá trình xử lý ảnh được chia làm các giai đoạn chính có thể mô tả ở hình 1

Một số hệ thống phân tích xử lý ảnh thu nhận qua camera như là con mắt của hệ thống thu nhận hình ảnh đầu vào. Có 2 loại camera: camera ống loại CCD và camera CCIR. Loại camera với tiêu chuẩn CCIR hình ảnh được quét với một tần số 1/25 và mỗi ảnh gồm 625 dòng... Loại CCD gồm các photo điốt có cường độ tương ứng sáng tại một điểm ảnh ứng với đó là một phần tử ảnh (pixel). Như vậy, ảnh là tập hợp các điểm ảnh. Độ phân giải (resolution) được gọi là số pixel tạo nên một ảnh

### 2.1.2. Một số khái niệm

- (*Picture Element*): phần tử ảnh

Trong xử lý ảnh thực tế là ảnh liên tục về giá trị cường độ sáng và về không gian. Để có thể xử lý nhiều tập ảnh bằng máy tính trước tiên cần thiết phải tiến hành số hóa ảnh. Trong quá trình ảnh được số hóa các tín hiệu được thay đổi liên tục sang tín hiệu rời rạc thông qua các quá trình thu thập lấy mẫu (rời rạc hóa về không gian), lượng hóa các thành phần thay đổi giá trị về nguyên tắc bằng mắt thường chúng ta không thể phân biệt được hai điểm kề nhau. Khái niệm *Picture element* là trong quá trình phần tử ảnh người ta áp dụng

Pixel hay đề cập đến trong các hệ thống xử lý đồ họa máy tính. Pixel thiết bị là tên gọi để tránh nhầm lẫn chúng ta tạm gọi khái niệm pixel. Pixel thiết bị có thể được nhận xét như sau: (trong chế độ đồ họa) khi ta quan sát màn hình, màn hình hiển thị điểm ảnh không liên tục mà gồm nhiều điểm nhỏ được hiển thị gọi là pixel. Mỗi pixel gồm một cặp tọa độ x, y và màu.

Cặp tọa độ x, y tạo nên *độ phân giải* (resolution). Như màn hình máy tính được sử dụng hiện nay có rất nhiều loại và với độ phân giải khác nhau: màn hình CGA có độ phân giải bình thường là 320x200; màn hình VGA là 640x350...

Vậy, một ảnh là một tập hợp các điểm ảnh. Đối với tập ảnh được số hóa, thường được hiển thị hoặc biểu diễn bởi nhiều bảng hai chiều  $I(n, p)$ :  $n$  dòng và  $p$  cột. Ta nói ảnh gồm  $n \times p$  pixels. Người ta thường ký hiệu  $I(x, y)$  để chỉ một pixel. Thường giá trị của  $n$  chọn bằng  $p$  và bằng 256.

- *Mức xám (Gray level):*

Là kết quả của sự mã hóa tương ứng một cường độ sáng của mỗi điểm ảnh tương ứng một hoặc nhiều giá trị số từ đó kết quả đạt được của quá trình lượng hóa. Cách mã hóa kinh điển thường dùng 16, 32 hay 64 mức. Mã hóa 256 mức là phổ dụng nhất do lý do kỹ thuật. Mỗi pixel sẽ được mã hóa bởi 8 bit. Vì  $2^8 = 256$  (0, 1, ..., 255), với 256 mức

- *Biểu diễn ảnh xám:*

Là Ảnh hai chiều, Giá trị được mô tả mức xám cho điểm ảnh trên cùng mặt phẳng mỗi điểm  $(x, y)$  trên ảnh ứng với một  $f(x, y)$ . Trong đó,  $x$  là số hiệu cột,  $y$  là số hiệu dòng trên ảnh.

Nếu là ảnh màu, có bảng màu thì  $f(x, y)$  là số hiệu màu.

Các giá trị R, G và B của điểm ảnh là thể hiện của Ảnh hơn 256 màu thì  $f(x, y)$  là 3 byte

- *Tách ngưỡng:*

Tách ngưỡng được hiểu như sau Giả thuyết ta có ảnh  $I$  ~ kích thước ảnh được thể hiện  $m \times n$ , gồm có hai số Min, Max và ngưỡng  $\theta$  khi đó:

Kỹ thuật tách ngưỡng thể hiện

```
for (i = 0; i < m; i++)
```

```
for (j = 0; j < n; j++)
```

```
I[i, j] = I[i, j] >=  $\theta$ ? Max: Min;
```

\* Áp dụng: Nếu Min = 0, Max = 1 sử dụng kỹ thuật chuyển một ảnh thành ảnh đen trắng được ta ứng dụng khi nhận dạng văn bản và khi quét có thể xảy ra ít sai sót nên thành ảnh hoặc ngược lại dẫn đến ảnh bị đứt nét hoặc dính không thể hiện rõ.

### 2.1.3. Các vấn đề cơ bản trong xử lý ảnh

**Chỉnh biến dạng:** Tập ảnh ta thu được có thể bị biến dạng, không được rõ nhiều do các thiết bị quang học và điện tử. Ta có thể khắc phục khi sử dụng các tập hợp phép chiếu được xây dựng dựa trên tập hợp các điểm điều khiển.

**Khử nhiễu:** Trong quá trình thu nhận ảnh thường có 2 loại nhiễu mà chúng ta cần loại bỏ: [5]

Hệ thống bị nhiễu: là nhiễu có quy luật ta có thể khử bằng cách áp dụng các phép biến đổi

Nhiễu ngẫu nhiên: là nhiễu không rõ nguyên nhân ta có thể khắc phục bằng nhiều phép lọc.

**Chỉnh số mức xám:** Chỉnh số mức xám là khắc phục tính không đồng đều của hệ thống thu thập xử lý hình ảnh, thông thường có 2 hướng tiếp cận được sử dụng nhiều:

**Giảm số mức xám:** Ta có thể thực hiện bằng cách nhóm nhiều mức xám gần nhau thành một nhóm. Thông thường các trường hợp ta giảm xuống hai mức xám thì chính là chuyển về ảnh đen trắng.

**Số mức xám tăng:** Sử dụng kỹ thuật nội suy ra các mức xám trung gian. Kỹ thuật này tăng cường độ mịn cho ảnh.

**Kỹ thuật Nén ảnh:** Là phương pháp nhằm giảm thiểu không gian lưu trữ, tiết kiệm bộ nhớ. Ở đây ta có hai hướng tiếp cận chính là ảnh được nén có bảo toàn và không được bảo toàn thông tin. Thực hiện nén không bảo toàn thì thường có khả năng nén cao hơn nhưng khi phục hồi được ảnh gốc là rất khó, ngược lại nén có bảo toàn cho phép khôi phục tối đa hoàn toàn ảnh gốc.

### 2.1.4. Nhận diện và phân loại ảnh

Nhận diện, Phát hiện tự động và truy vết, phân loại và nhóm các mẫu là những vấn đề rất quan trọng của thị giác máy tính, Khi mà thời đại công nghệ ngày càng phát triển được ứng dụng trong nhiều ngành, lĩnh vực khác nhau. Ví dụ như phát hiện dấu vân tay, phân loại hàng hóa, phát hiện khuôn mặt hay một biểu đồ của âm thanh. Để phát hiện hoặc phân loại được ảnh có thể sử dụng hai cách chính:

- **Học có giám sát** (supervised learning): Phân loại mẫu đầu vào vào một lớp đã xác định.
- **Học không giám sát** (unsupervised learning): Các mẫu được gom nhóm với nhau dựa trên một số điều kiện nào đó. Tập hợp các lớp ảnh này cho đến thời điểm ta xử lý phân loại vẫn chưa biết hay chưa được định danh.

## 2.2. Giới thiệu về OpenCV

### 2.2.1. Tổng quan về OpenCV

OpenCV (Open Source Computer Vision Library) là thư viện mã nguồn mở hàng đầu cho thị giác máy tính (computer vision), Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực. Máy học, xử lý hình ảnh và các tính năng tăng tốc tối ưu GPU trong thời gian thực.

OpenCV được phát hành theo giấy phép BSD, Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV còn có các interface Python, Java, C++, C, và hỗ trợ, Windows, Linux, Mac OS, iOS và Android. OpenCV được xây dựng thiết kế để hỗ trợ các phép tính toán hiệu quả và với sự tập trung chính vào các ứng dụng thời gian thực. Được tối ưu hóa C/C++, ta có thể tận dụng thư viện xử lý đa lõi. Được sử dụng rộng rãi trên toàn thế giới, Phạm vi sử dụng phổ biến trong rất nhiều trong tương tác lĩnh vực nghệ thuật, khai thác mỏ, cho đến lĩnh vực bản đồ trên web hoặc công nghệ robot.



### 2.2.2. Cấu trúc OpenCV

Sơ lược về cấu trúc của thư viện OpenCV: 2 phần (module) chính trong thư viện OpenCV. Mã nguồn được nhóm phát triển xây dựng và kiểm định toàn diện, gồm nhiều thuật toán đã được thế giới công nhận và đánh giá dựa trên cơ sở lý thuyết chắc chắn là Phần căn bản (basic). Các tổ chức khoa học khác nhau trên thế giới đóng góp xây dựng các thuật toán cập nhật được xây dựng từ nhiều công trình nghiên cứu là Phần mở rộng (contribution) được các bài báo mới đăng trong thời gian gần đây. Từ phiên bản 3.0, phần mở rộng được tách riêng không còn được gộp chung với thư viện mặc định. Nhiều thuật toán được nghiên cứu xây dựng trong phần mở rộng được nâng cấp bổ sung có độ ổn định và tối ưu không cao. OpenCV là thư viện mã nguồn mở được đóng gói thành tập tin nén. Tùy vào hệ điều hành khác nhau mà tập tin nén này có định dạng tương ứng. Thư viện OpenCV cung cấp cho người dùng từ các cấu trúc dữ liệu, các hàm các đối tượng khai báo nguyên mẫu (prototype) của chúng trong các tập tin thư viện C/C++ (\*.h, \*.hpp,...) và được định nghĩa chi tiết bởi các tập tin mã nguồn (\*.c, \*.cpp). Với mức độ sử dụng OpenCV, cài đặt các tập tin đã được biên dịch sẵn rồi thực hiện các thao tác cài đặt đường dẫn cho thích hợp để chương trình tìm đến đúng vị trí của các tập tin thư viện. Ở mức độ cao hơn, ta phải biên dịch lại mã nguồn trực tiếp trên máy trước khi cài đặt nếu ta muốn hiệu chỉnh cập nhật sửa đổi thuật toán hay sử dụng các phần mở rộng của OpenCV.

### 2.2.3. Các ứng dụng OpenCV

OpenCV đang được ứng dụng rộng trong các ứng dụng bao gồm:

- Hình ảnh street view
- Kiểm tra và giám sát tự động
- Robot và xe hơi tự lái
- Phân tích hình ảnh y tế
- Tìm kiếm và phục hồi hình ảnh/video
- Phim - cấu trúc 3D từ chuyển động
- Nghệ thuật sắp đặt tương tác

#### 2.2.4. Chức năng OpenCV

- Image/video I/O, xử lý, hiển thị (core, imgproc, highgui)
- Phát hiện các vật thể (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu).

### 2.3. Các Phương pháp phát hiện chuyển động (Sử dụng các kĩ thuật trù nền trong camera giám sát)

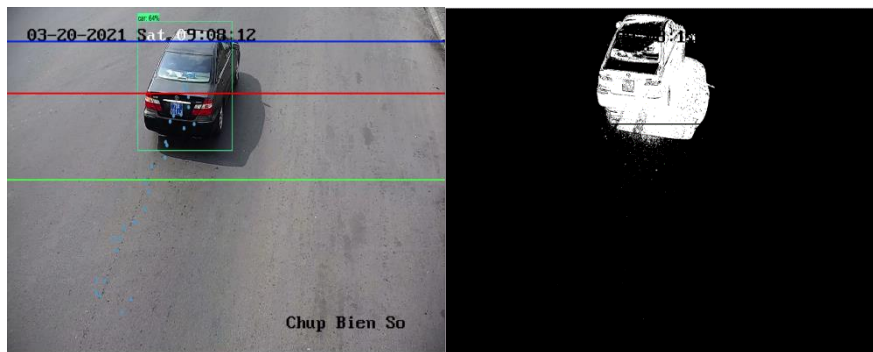
#### 2.3.1. Phát hiện đối tượng chuyển động là gì ?

Phát hiện đối tượng chuyển động [6] thực ra là quá trình phân tích xử lý đưa ra nhận biết các vết đối tượng chuyển động từ nhiều khung hình video. Quá trình này là quá trình xử lý tập hợp các chuỗi ảnh liên tiếp trong một đoạn video từ đó ta có thể phát hiện nhận dạng các đối tượng chuyển động trong một đoạn khung hình ảnh.



Hình 2: Các quá trình phát hiện chuyển động

### 2.3.2. Phát hiện các vùng ảnh nổi



**Hình 3: Phát hiện vùng ảnh nổi**

Phát hiện phân loại các vùng ảnh nổi là một trong những module thiết yếu trong hệ thống phát hiện truy vết theo đối tượng chuyển động qua khung hình video. Module này đóng vai trò quan trọng phân tích xử lý các khung hình video để lọc ra các vùng ảnh nổi lên trên tách biệt với phần còn lại của các khung hình nền. Phương pháp trừ ảnh nền (Background Subtraction)

- Phương pháp trừ ảnh nền (Background Subtraction)

Là phương pháp so sánh ảnh đơn giản nhất. Background Subtraction là phương pháp dựa trên sự sai khác giữa hai hoặc nhiều ảnh và tiến hành so sánh sự sai khác này với một hoặc nhiều giá trị ngưỡng cho trước. Trường hợp mà sự sai khác này lớn hơn hoặc nhỏ hơn giá trị ngưỡng đã cho, thì ở đây có nghĩa là đã có sự sai khác về các giá trị pixel ảnh. Ta đưa ra kết luận có đối tượng chuyển động.

- Phương pháp dựa trên thống kê (Statistical Methods)

Phương pháp dựa trên thống kê các đặc điểm riêng của các điểm ảnh từ đó tính toán để phát triển các thuật toán có thể khắc phục vượt qua được những khó khăn vướng mắc mà phương pháp trừ nền chưa xử lý được đó là những khó khăn gặp phải khi khung cảnh luôn bị tác động thay đổi bởi nhiều yếu tố môi trường.

Những tập hợp Các điểm ảnh nổi sẽ được phân tích xác định bởi việc so sánh các thông tin của nhiều điểm ảnh với các mô hình ảnh nền. Phương pháp dựa trên thống kê này được lựa chọn áp dụng khá phổ biến trong những khung cảnh, khung hình có nhiều nhiễu, ánh sáng thay đổi liên tục bởi môi trường và các yếu tố khác.

Phương pháp thống kê cơ bản vẫn phải áp dụng dựa trên kỹ thuật trừ ảnh nền song được sử dụng thêm vào đó nó còn được thực hiện ăn khớp, và cập nhật động thông tin lại các điểm ảnh thuộc ảnh nền đang được xử lý

Phương pháp này được đưa ra dựa trên sự chênh lệch tạm thời giữa tập hợp các khung hình dựa trên sự chênh lệch tạm thời giữa các khung hình Phương pháp này phát hiện ra những vùng thay đổi chuyển động bằng cách sử dụng các thuật toán so sánh sự khác nhau ở mức điểm ảnh giữa hai hoặc nhiều khung hình liên tiếp trong một chuỗi các khung hình video. Sử dụng phương pháp này là khá tối ưu tốt đối với khung cảnh có nhiều chuyển động động nhiều thay đổi song nó lại chưa được chính xác trong việc phát hiện ra các điểm ảnh có liên quan đến một số kiểu chuyển động.

### ***2.3.3. Xử lý các vùng ảnh nổi***

Xử lý các vùng ảnh nổi trên thực tế vẫn còn nhiều khó khăn do nhiều do đó cần để phải lọc nhiễu, xử lý ghép nhiều vùng phân mảnh, phân tích loại bỏ các vết không liên quan thì cần phải áp dụng một số module xử lý các vùng ảnh nổi để đưa ra sản phẩm là các đối tượng chuyển động rõ ràng để từ đó ta mới xác định được tính chất, đặc trưng của chúng.

### ***2.3.4. Các kỹ thuật trừ ảnh***

Hiện nay, có nhiều nghiên cứu và kỹ thuật xử lý ảnh, trừ ảnh [6] và có thể chia thành 5 loại như sau:

- ✓ Trừ ảnh dựa vào khối.
- ✓ Trừ ảnh dựa vào biểu đồ.
- ✓ Trừ ảnh dựa vào điểm ảnh.
- ✓ Trừ ảnh dựa vào thống kê.
- ✓ Trừ ảnh dựa vào đặc trưng

- ***Trừ ảnh dựa vào khối***

Trừ ảnh dựa vào khối lượng tiếp cận phân khối sử dụng các tài nguyên có đặc tính cục bộ nhằm tách biệt tăng tính độc lập với các di chuyển của camera và đối

tượng. Mỗi khung hình được chia thành  $b$  khối. Tập hợp các khối trên khung hình  $f_1$  được đưa vào so sánh với khối tương ứng với trên khung hình  $f_2$ . Về cơ bản thì độ chênh lệch giữa nhiều khung hình được tính như sau:

$$D(f_1, f_2) = \sum_{k=1}^b C_k \cdot DP(f_1, f_2, k) \quad (1)$$

$C_k$  là hệ số được cho trước.

$DP(f_1, f_2, k)$  độ chênh lệch giữa nhiều khối thứ  $k$  của nhiều khung hình  $f_1$  và  $f_2$ .

Một hướng tiếp cận khác so với kỹ thuật trừ ảnh phân khối được Shahraray cho rằng chia khung hình thành 12 miền và tìm những miền tối ưu thích hợp nhất cho mỗi miền ở khung hình kia. Sử dụng Độ chênh lệch tính bằng kỹ thuật trừ ảnh dựa vào các điểm ảnh thời điểm hiện tại của từng miền được sắp xếp. Từ đó ta tính tổng có trọng số được sắp xếp của các chênh lệch cho ta kết quả  $D$  cuối cùng.

- ***Trừ ảnh dựa vào biểu đồ***

Ta có thể áp dụng biểu đồ biểu đồ màu hoặc mức xám để tính toán sự sai khác giữa hai khung hình vì sự phân bố màu trên các frame liên tục thường thì không bị ảnh hưởng nhiều bởi chuyển động của chuyển động của đối tượng và camera.

Biểu đồ màu hay còn được gọi là mức xám của khung hình  $I$  là vector  $G$  chiều  $H_i = (H_i(1), H_i(2), H_i(3), \dots, H_i(G))$ .

Trong đó:

$G$  là số màu (mức xám)

$H_i(j)$  là số điểm ảnh của khung hình  $i$  có màu (mức xám) $j$ .

Biểu đồ toàn cục hoặc biểu đồ cục bộ là hai phương pháp chính của Phương pháp trừ ảnh dựa trên biểu đồ.

- Biểu đồ được hiển thị biểu diễn sự phân bố giá trị màu (mức xám) của toàn bộ khung hình được gọi là Biểu đồ toàn cục.

- Biểu đồ chỉ mô tả lại sự phân bố màu hay mức xám hiển thị của phần nào đó của khung hình là Biểu đồ cục bộ.

- **Trừ ảnh dựa vào điểm ảnh**

Phương pháp để tính toán sự sai khác giữa hai frame bằng việc tính toán một tập hợp các giá trị, nó mô tả mọi lại toàn bộ thay đổi về cường độ điểm ảnh trong các ảnh hiển thị trên khung hình. Có nhiều phương pháp được áp dụng để tính sự sai khác nhau này, Nagasaka và tanaka đã tìm ra phương pháp tối ưu để xử lý vấn đề là phương pháp tính tổng toàn bộ những thay đổi khác nhau về cường độ điểm ảnh được phân tích giữa hai khung hình như nhau là độ chênh lệch khung  $D(f_1, f_2)$ .

$$D(f_1, f_2) = \frac{1}{X \times Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} |f_1(x, y) - f_2(x, y)| \quad (2)$$

- **Trừ ảnh dựa vào thống kê**

Ở phương pháp này ta dựa vào phương pháp trừ giá trị điểm ảnh. Ta không đi tính tổng sự sai khác của tất cả các điểm ảnh, ta dùng các thuật toán chia nhỏ ảnh thành tập các miền rồi so sánh các đại lượng thống kê trên các điểm ảnh của các miền đó.

Trên toàn bộ khung hình biểu diễn hiển thị sử dụng một giá trị Ta sử dụng thống kê tỷ lệ số tập hợp điểm ảnh thay đổi khác nhau

d: được tính giữa hai điểm ảnh tương ứng Là ngưỡng sai khác.

Gọi S là tập các điểm ảnh có độ sai khác lớn hơn d:

$$S = \{(x, y) \mid |f_1(x, y) - f_2(x, y)| > d\} \quad (3)$$

Tính độ sai khác giữa các khung hình được tính bằng tỷ lệ từ các điểm ảnh nhận được có độ chênh lệch lớn hơn d.

$$D(f_1, f_2) = \frac{S.count}{X * Y} \quad (4)$$

Các khác dùng đúng các đại lượng thống kê cho từng miền, nhu biểu đồ chẳng hạn. Phương pháp này còn khá nhiều sai sót trong phát hiện cảnh phim

**Đặc trưng là vector chuyển động:**

Người ta thường thấy các hiệu ứng do những chuyển động của các đối tượng trong camera, như pan (quét), zoom (zoom in - phóng to, zoom out - thu nhỏ), italic (nghiêng).

***Đặc trưng là cạnh:***

Phân loại và phát hiện chuyển cảnh là một tập hợp các sự phát hiện sự xuất hiện các cảnh trong một khung hình, chúng cách các cảnh trong nhiều khung hình trước một khoảng kích thước nhất định. Kỹ thuật này phát hiện và phân loại các loại chuyển cảnh: chồng mờ, Cắt cứng, fade, wipe.

**2.3.5. Một số kỹ thuật trừ nền cơ bản**

- *Sự khác biệt khung (Frame Differencing)*

Trong nhiều phương pháp trừ nền thì phương pháp khác biệt khung được xem là phương pháp đơn giản nhất. Tốc độ thực thi thuật toán nhanh, Chi phí tính toán thấp,.. Tuy nhiên kết quả đạt được khi ta áp dụng chạy thuật toán này là tương đối thấp.

Các đối tượng chuyển động sẽ được phân tích xử lý phát hiện dựa trên sự khác biệt giữa hai khung ảnh liên tiếp nhau cùng với một ngưỡng được chọn trước là Ý tưởng chính trong phương pháp khác biệt khung. Từ đó việc sử dụng Giải thuật trên đối với mỗi giá trị pixel kết quả ta so sánh giá trị tại pixel đó với ngưỡng đã được chọn được thực hiện bằng phương pháp trừ hai khung liên tiếp. Các tập hợp giá trị này nằm trong một ngưỡng cho trước được cho phép thì tại đó ta xem như là nền. Ngược lại không phải là nền.

- *Lọc trung bình (Running Average)*

Phương pháp trừ nền trung bình được phát triển cải tiến của phương pháp trừ nền cơ bản. Từ những cái cơ bản thay vì giữ nguyên giá trị khung hình nền  $B(x,y)$  trong phép trừ nền, phương pháp trừ nền trung bình được phát triển cải tiến bằng việc cập nhật liên tục các giá trị lọc tách khung hình nền nhằm tăng tính chính xác cho kết quả phát hiện nhận diện đặc trưng của đối tượng chuyển động.

**Thuật toán tổng quát đối với pixel  $(x,y)$ :**

$\tau$ : Tham số ngưỡng cho trước

$\beta$ : Giá trị cập nhật nền cho trước  $\in [0,1]$

$B_t(x,y)$ : Giá trị khung nền

$I_t(x,y)$ : Giá trị khung video đến

Giá trị khởi tạo mô hình nền và frame video đến:  $B_0(x,y) = I_0(x,y)$

Bước 1: Cập nhật giá trị nền  $B_0(x,y)$  theo chỉ số frame t:

$$B_t(x,y) = (1 - \beta)B_{t-1}(x,y) + \beta I_t(x,y) \quad (5)$$

Bước 2: Tính mặt nạ nhị phân phát hiện đối tượng chuyển động  $D(x,y)$ :

$$D(x,y) = \begin{cases} 1, & \text{if } |I_t(x,y) - B_t(x,y)| > \tau \\ 0, & \text{if } |I_t(x,y) - B_t(x,y)| \leq \tau \end{cases} \quad (6)$$

$D(x,y) = 0$  (Ngưỡng cho trước  $\tau$  là tính hiệu hai khung hình nhỏ hơn giá trị) đưa ra kết luận ở đây là những pixel của nền, ngược lại.

$D(x,y) = 1$  (Ngưỡng  $\tau$  cho trước tính hiệu hai khung hình lớn hơn) nhận xét đưa ra kết luận đây là những pixel của đối tượng di chuyển. Tập hợp các pixel này ta sẽ tính được hình ảnh của đối tượng chuyển động

#### **Đánh giá:**

- Các giá trị được cập nhật liên tục những giá nền thông qua việc sử dụng tham số  $\beta$  đã làm cho thuật toán tăng cường độ chính xác hơn cho việc phát hiện đối tượng chuyển động sau đó ta so với thuật toán trừ nền cơ bản.

- Cài đặt giải thuật đơn giản giảm thiểu thời gian chạy, độ phức tạp tính toán đơn giản, tăng tốc độ xử lý nhanh.

#### **Nhược điểm:**

Phương pháp này còn nhiều hạn chế chưa giải quyết được:

- Không thể phát hiện được sự thay đổi lớn trong một vùng ảnh nhỏ và ngược lại, camera thu ảnh có còn bị nhiễu.

- Kết quả phát hiện chưa thật chính xác đối với những trường hợp đối tượng trong đoạn video không di chuyển.

- Cũng như chưa khắc phục được những hạn chế của phương pháp trừ nền cơ bản.



### 2.3.6. Kỹ thuật trừ nền nâng cao

#### Mô hình Gauss hỗn hợp (Mô hình GMM)

Stauffer and Grimson (1999) đã đưa ra phương pháp trộn các mô hình nền lai để giải quyết vấn đề ánh sáng thay đổi, sự lộn xộn từ khung cảnh thực tế các hành động lặp lại. Chứng minh một mô hình nền đơn thì không thể xử lý được các khung hình liên tục trong một thời gian dài.

Sử dụng phương pháp pha trộn phân tán Gauss ta có thể dùng để biểu diễn mỗi điểm ảnh trên một mô hình.

Theo luận điểm trên, thực hiện và tích hợp sử dụng phương pháp này vào trong hệ thống giám sát. Trong mô hình này, ta coi các giá trị của điểm ảnh theo thời gian xem như là một tiến trình điểm. Tiến trình điểm là một tập hợp chuỗi giá trị điểm theo thời gian, nghĩa là giá trị cấp xám hoặc là véc tơ đối với ảnh màu.

Tại thời điểm  $t$ , tại điểm  $(x_0, y_0)$  thì  $\{X_1, \dots, X_t\} = \{I(x_0, y_0, i), 1 \leq i \leq t\}$  là một tiến trình điểm. Đây là một trong những yếu tố hướng dẫn cách lựa chọn mô hình và các thủ tục cập nhật. Dãy  $\{X_1, \dots, X_t\}$  được mô hình hóa bởi  $K$  phân bố Gauss. Tần suất thực hiện quan sát của điểm ảnh hiện tại được tính bởi công thức:

$$P(X_t) = \sum_{i=1}^k \omega_{i,t} \eta(X_t, \mu_i, \Sigma_{i,t}) \quad (7)$$

Trong đó:

$\omega_{i,t}$  là trọng số tương ứng với Gauss thứ  $i$  tại thời điểm  $t$ ;

$\mu_{i,t}$  là giá trị tính trung bình các Gauss thứ  $i$  tại thời điểm  $t$ ;

$\Sigma_{i,t}$  là tập ma trận hiệp phương sai của phân bố Gauss thứ  $i$  tại thời gian  $t$ ;

$\eta$  là hàm mật độ xác suất được xác định bởi công thức:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (8)$$

$K$  phụ thuộc vào bộ nhớ và khả năng xử lý của máy tính

Ma trận hiệp phương sai được giả thiết có dạng sau:

$$\Sigma_{k,t} = \sigma_k^2 I \quad (9)$$

$I$ : là ma trận đơn vị.

R, G, B giá trị của các điểm ảnh là các thành phần được cho là độc lập và có cùng một phương sai. Thủ tục phát hiện các điểm tiền cảnh:

Ta phân phối K-Gauss cho một điểm ảnh được khởi tạo với các giá trị trung bình, với giá trị trọng số thấp như các công thức ở trên.

Khi có một điểm ảnh mới trong chuỗi khung hình được quan sát, để xác định kiểu của nó,

Vector RGB được so sánh kiểm tra lại so với K-Gauss cho đến khi tìm ra được trường hợp đúng.

Giá trị điểm ảnh trong vòng 2.5 độ lệch chuẩn của một phân phối được xác định khi giá trị tìm được phù hợp.

Tiếp theo, xét K phân phối tại thời điểm

t:  $\omega_{i,t}$  được cập nhật theo công thức:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha M_{k,t} \quad (10)$$

$\alpha$  là tỉ lệ học,  $M_{k,t=1}$  với phân phối Gauss và  $M_{k,t=0}$  đối với các phân phối khác.

Tiếp theo sau khi bước trên kết thúc, các trọng số phân phối được chuẩn hóa và các tham số nhằm xác định chính xác hóa Gauss sẽ được cập nhật với các quan sát tiếp theo:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (11)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^t(X_t - \mu_t) \quad (12)$$

Trong đó:

$$\rho = \alpha\eta(X_t, \mu_t, \sigma_k)$$

Trường hợp không đúng thì phân phối Gauss với những kết quả có xác suất thấp nhất sẽ được thay thế bởi xác suất phân phối mới với giá trị khác là giá trị điểm ảnh mới được thêm vào.

Bài toán phát hiện ra loại điểm ảnh (tiền cảnh hay nền), phân phối K-Gauss được sắp xếp bởi giá trị  $\omega/\rho$ .

Danh sách này các tập hợp điểm ảnh thuộc ảnh nền sẽ tương ứng với xác suất phân phối thuộc trọng số lớn và ít biến đổi. B sẽ được phân phối như là một mô hình xác định ảnh nền theo công thức:

$$B = \arg \min_b \left\{ \sum_{i=1}^b \omega_{i,t} \mid \sum_{i=1}^b \omega_{i,t} > T \right\} \quad (13)$$

T: Giá trị nhỏ nhất được tìm trong mô hình ảnh nền

## 2.4. Phương pháp theo vết đối tượng

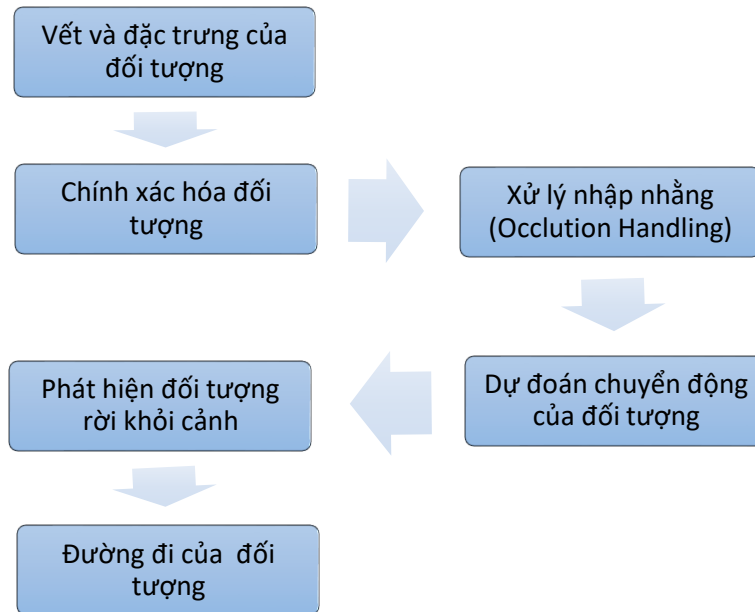
### 2.4.1. Theo vết đối tượng

Theo vết đối tượng [7] là các quá trình chính xác hóa đối tượng trong các khung hình liên tiếp để từ đó ta đưa ra các thông tin về chuyển động của đối tượng như loại xe, đường đi, tốc độ và hướng chuyển động.

Theo vết đối tượng bao gồm hai giai đoạn:

- Cô lập những đối tượng ra khỏi nền trong từng frame
- Sự liên kết của các đối tượng trong các frame kế tiếp để theo vết.

Mục tiêu của phương pháp theo vết đối tượng là tìm ra đối tượng chuyển động đã được phát hiện trong giai đoạn như hình 4, ta có thể phát hiện đối tượng chuyển động từ khung hình thứ nhất sang khung hình thứ hai trong một dãy tập hợp các khung hình.



**Hình 4: Quy trình theo vết đối tượng**

Phần quan trọng nhất trong hệ thống là theo vết đối tượng chuyển động. Kết quả phát hiện đối tượng chuyển động sẽ ảnh hưởng trực tiếp đến tốc độ xử lý và độ chính xác của toàn hệ thống. Nếu kết quả phát hiện đối tượng chính xác, hệ thống sẽ xử lý nhanh và đạt được độ chính xác cao và ngược lại.

#### **2.4.2. Quy trình theo vết đối tượng**

Việc ứng dụng hệ thống theo vết trong hệ thống camera giám sát thông minh còn gặp nhiều khó khăn do một số nguyên nhân sau đây:

- Tốc độ xử lý dữ liệu đường truyền, xử lý trong thời gian thực.
- Cách tiếp cận theo vết đối tượng dựa dựa vào như màu sắc của đối tượng. Cách tiếp cận này sử dụng mô hình Kalman đã được phát triển trong giao tiếp giám sát bằng máy tính.
- Truy vết đối tượng dựa vào các đường bao động (*Active Contour*): Cách tiếp cận ta dựa trên biên bao gồm tìm ra các đường biên bao đặc trưng của một đối tượng và liên tục cập nhật tự động lại các thuộc tính của đối tượng đó.
- Ảnh nền lộn xộn: Do có sự có mặt của nhiều đối tượng khác, sự chuyển động hỗn loạn trong khung hình.

- Cường độ chiếu sáng thay đổi: Thay đổi hướng và cường độ chiếu sáng
- Thay đổi điểm nhìn: Thay đổi vị trí của đối tượng.

Nếu đối tượng chỉ có một vài đặc trưng thị giác, như màu sắc.... dùng để biểu diễn thì việc xác định tất cả các điểm ảnh cùng màu với đối tượng khá dễ dàng. Nhưng trên thực tế, còn phụ thuộc vào nhiều yếu tố khác như đối tượng phát hiện, số lượng và theo vết như nào. Đối tượng trong video hầu hết đều chuyển động, khi xuất hiện nhiều đối tượng vào cùng một vùng giám sát việc theo vết rất khó trong việc xác định vết nào của đối tượng nào như: cường độ chiếu sáng tối khác nhau, góc quay sẽ làm cho việc phát hiện, nhận dạng và theo dõi sẽ gặp rất nhiều khó khăn.

Trong một đối tượng có rất nhiều đặc trưng: đặc trưng về cấu trúc, đặc trưng về mô hình, đặc trưng về màu sắc. Do các đối tượng có nhiều đặc trưng Vì vậy cũng có rất nhiều phương pháp theo vết đối tượng khác nhau như: theo vết đối tượng theo hướng hình học, theo đặc trưng của đối tượng. Để giải quyết vấn đề tổng quát lại ta có thể chia thành bốn cách tiếp cận chính như:

Theo vết dựa trên mô hình: Hệ thống dựa vào mô hình học cấu trúc 2D hay 3D của đối tượng. Để khắc phục những nhược điểm trong mô hình 3D thì Các mô hình 2D được giới thiệu. Việc tìm ra đường đi của các mô hình yêu cầu độ chính xác cao mà đối tượng theo dõi là ít mô hình này được áp dụng hiệu quả.

Theo vết dựa trên tên miền: Ý tưởng là nhận dạng những miền liên kết với nhau trong ảnh, khối mà có liên kết với mục tiêu được theo dõi. Ưu điểm của cách tiếp cận này là: Các tham số của thuật toán cũng có ý nghĩa vật lý, dễ dàng khái niệm hóa. Khởi tạo khá dễ dàng, chỉ cần có vị trí và kích thước của cửa sổ cần được định nghĩa.

Theo vết đối tượng dựa vào đường bao động (*Active Contour*): Cách tiếp cận dựa trên biên bao gồm tìm đường biên bao của một đối tượng và liên tục cập nhật tự động đối tượng đó. Nơi khớp nhất sẽ là đường biên hiện tại, mô hình sẽ cập nhật đường biên hiện tại để phản ánh hình dáng của đối tượng trong frame hiện tại. Quá trình này sẽ được lặp đi lặp lại với mô hình đường biên được cập nhật. Ưu điểm của cách tiếp cận này là với tình huống đối tượng bị che khuất một phần có khả năng xử

lý hiệu quả. Tuy nhiên vấn đề cần quan tâm của mô hình là nó yêu cầu sự khởi tạo chính xác và xử lý như nào với trường hợp nhập nhằng.

Cách tiếp cận này sử dụng mô hình Kalman đã được phát triển trong giao tiếp giám sát bằng máy tính. Chọn các đặc trưng của đối tượng và theo vết dựa trên những đặc trưng đó, ví dụ ta theo vết bằng việc theo dõi một nhóm tập hợp điểm trên một góc nào đó của đối tượng, trong các frame ta tìm đối tượng bằng cách tìm tập hợp các điểm đó sao cho giống nhất. Ưu điểm của phương pháp này là có thể giải quyết được sự che khuất của đối tượng, khi một số đặc trưng của đối tượng vẫn còn thì ta vẫn có thể tiến hành theo vết được. Tuy nhiên chính vì việc theo vết dựa trên đặc trưng của đối tượng nên yêu cầu những đặc trưng này là duy nhất cho đối tượng cần theo dõi, không được phổ biến, lẫn với các đối tượng khác. Việc chọn các đặc trưng này khó và tốn nhiều thời gian.

### ***2.4.3. Các Phương pháp dựa trên Deep Learning***

Một số kỹ thuật phát hiện và nhận dạng đối tượng dựa trên mạng CNN phổ biến trên thế giới như: R-CNN(Region-Convolutional Neural Network)[16], Fast R-CNN[17], Faster R-CNN[18], SSD (Single Shot Multibox Detector)[19] ...

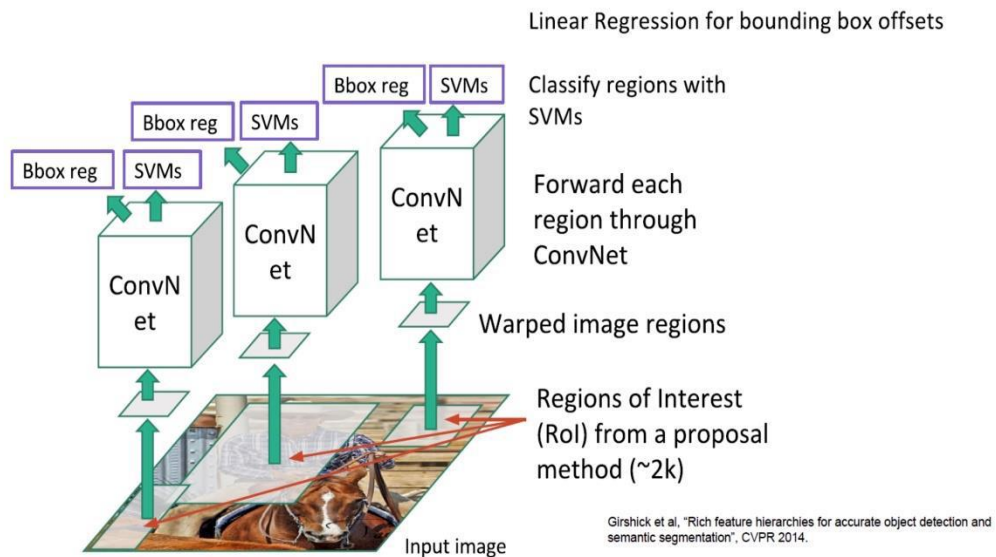
- ***R-CNN***

Các nghiên cứu về các kỹ thuật học sâu các kỹ thuật phân loại dựa trên HOG[19] ta thu được nhiều kết quả rõ ràng dần được cải tiến thay thế bằng một kỹ thuật học sâu như CNN đã cho kết quả chính xác hơn. Tuy nhiên, có một vấn đề là CNN quá chậm và tính toán rất tốn kém. Không thể chạy CNN trên nhiều cửa sổ được tạo bởi thuật toán cửa sổ trượt (sliding window detector). R-CNN[17] đã giải quyết vấn đề này bằng cách chạy một thuật toán được gọi là Selective Search để giảm số hộp giới hạn (bounding box) được đưa vào bộ phân loại. Selective Search là sử dụng các dấu hiệu được bố cục như cường độ, kết cấu, màu sắc ... để tạo ra các vị trí có thể chứa đối tượng. CNN được dựa trên các phân loại mà chúng ta có thể cung cấp các hộp giới hạn.

Hãy nhớ rằng, lớp kết nối đầy đủ (Fully connected layer) của CNN có một đầu vào kích thước cố định vì vậy chúng ta thay đổi kích cỡ của tất cả các hộp giới hạn

về một kích thước cố định và cũng cấp cho CNN. Do đó R-CNN[17] thường có 3 phần quan trọng.

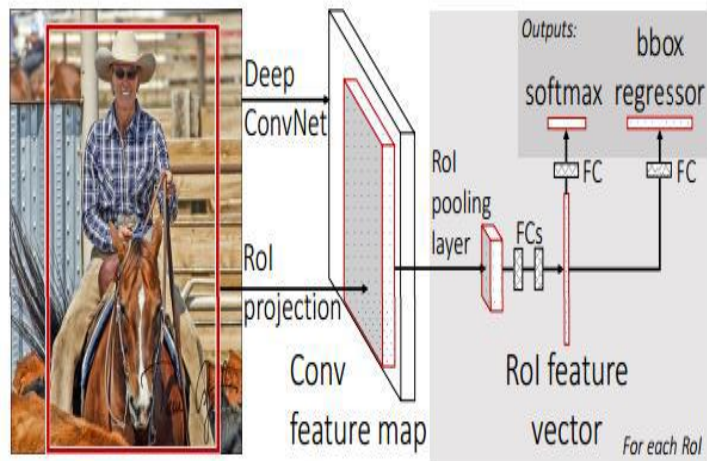
- Chạy Selective Search để tìm kiếm các hộp các đối tượng có thể xuất hiện
- Đưa các hộp đó vào CNN, sau đó là SVM[17] để dự đoán lớp của từng hộp
- Tối ưu hóa các hộp bằng cách huấn luyện



**Hình 5: Mô hình R-CNN**

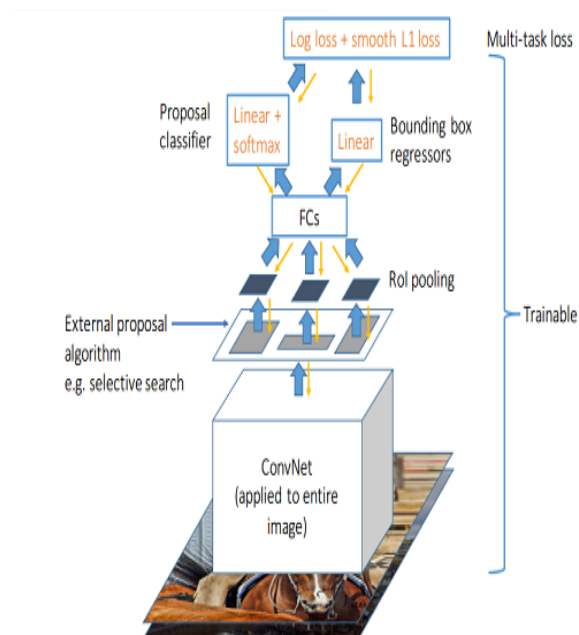
- **Fast R-CNN**

Tuy R-CNN[16] đã đạt được nhiều tiến bộ nhưng nó vẫn còn rất chậm. Bởi vì nó phải chạy CNN với 2000 khu vực được đề xuất bởi Selective Search dẫn đến mất rất nhiều thời gian. SPP-net[20] (Spatial Pyramid Pooling Network) đã cố gắng tối ưu nó. Với SPP-net các tính toán đặc trưng trên CNN chỉ thực hiện một lần và có thể sử dụng để tính toán các đặc trưng cho mỗi khu vực được tạo bởi Selective Search.



**Hình 6: Mô hình SPP-net**

Fast R-CNN[23] sử dụng ý tưởng của SPP-net, R-CNN và sửa một vấn đề chính trong SPP-net như: Fast R-CNN có thể thực hiện từ đầu đến cuối (end-to-end). Một điều nữa là họ đã thêm tính toán hồi quy để tìm hộp giới hạn vào việc huấn luyện. Vì vậy mạng có hai đầu ra một đầu phân loại đầu ra và một đầu dự đoán hộp giới hạn. Mục tiêu này là tính năng nổi bật của Fast R-CNN[17] vì nó không còn yêu cầu huấn luyện mạng độc lập để tìm vị trí và phân loại đối tượng. So với SPP-net hai thay đổi này đã tối ưu và làm giảm bớt thời gian huấn luyện tổng thể và tăng độ chính xác.



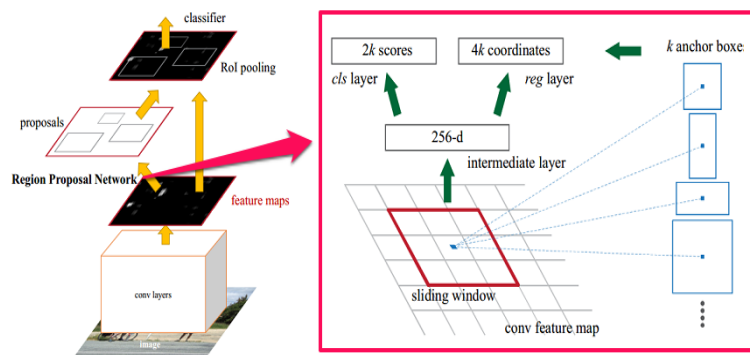
**Hình 7: Mô hình mạng Fast R-CNN**



- ***Faster R-CNN***

Faster R-CNN[18] là một phiên bản cải tiến của Fast R-CNN[17]. Mạng đề xuất khu vực đề xuất ra các vùng quan tâm trên ảnh được gọi là Region Proposal network-RPN

Selective search bằng một mạng CNN nhỏ Nó thay thế phần chậm nhất của Fast R-CNN.



**Hình 8: Mô hình mạng Faster R-CNN**

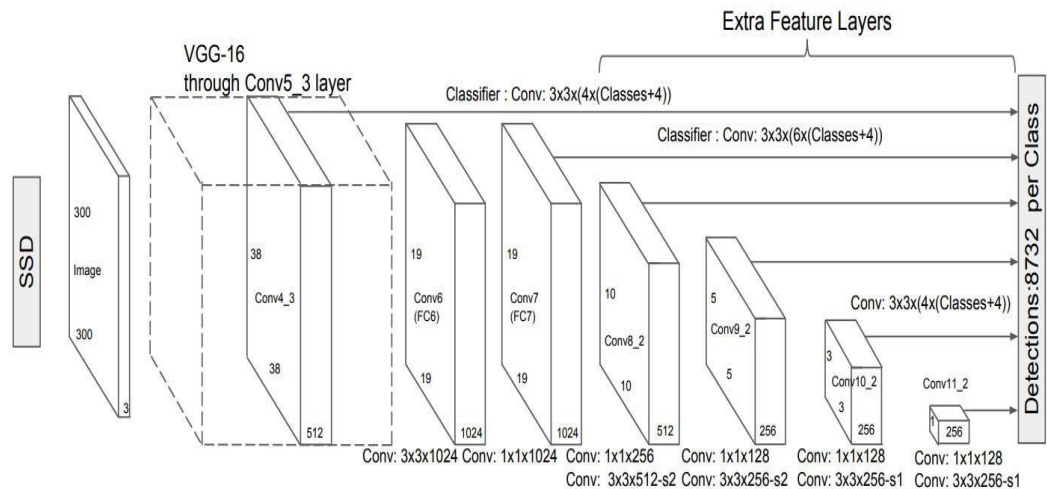
Để xử lý vấn đề các đối tượng có các tỉ lệ khác nhau, Faster R-CNN đã giới thiệu ý tưởng về các hộp neo (anchor box). Tại mỗi vị trí sẽ sử dụng 3 hộp neo có các kích thước 128x128, 256x256, 512x512. Tương tự với tỉ lệ 1:1, 2:1 và 1:2. Vì vậy, tổng cộng tại mỗi vị trí, chúng ta có 9 hộp mà RPN sẽ dự đoán xác suất xuất hiện của nó là nền hay đối tượng. Chúng ta áp dụng hồi quy cho hộp giới hạn để cải thiện độ chính xác của các hộp neo ở mỗi vị trí. Vì vậy, RPN đưa ra các hộp giới hạn có kích thước khác nhau với xác suất tương ứng mỗi lớp. Phần còn lại tương tự như Fast R-CNN. Những thay đổi trên giúp cho Faster R-CNN nhanh gấp 10 lần so với Fast R-CNN với độ chính xác tương tự trên tập dữ liệu như VOC-2007[120]. Đó là lý do tại sao Faster R-CNN là một trong những thuật toán phát hiện đối tượng chính xác nhất. Dưới đây là so sánh giữa các phiên bản R-CNN

**Bảng 1: So sánh các mô hình dựa trên R-CNN**

	R-CNN	Fast R-CNN	Faster R-CNN
Thời gian xử lý 1 ảnh	50 giây	2 giây	0.2 giây
Tốc độ tăng	1x	25x	250x

Các phương pháp được đề cập ở trên xử lý việc phát hiện đối tượng bằng cách tạo ra các đề xuất khu vực sau đó các đề xuất đó sẽ được chuyển qua một bộ phân loại để xác định đối tượng. Tuy nhiên, có một số phương pháp khác coi việc phát hiện đối tượng như một bài toán hồi quy SSD[19] là phương pháp phổ biến nhất.

- **Single Shot Detector (SSD)**

**Hình 9: Mô hình mạng SSD**

SSD[19] đạt được sự cân bằng tốt giữa tốc độ và độ chính xác. SSD chỉ chạy trên một mạng CNN trên hình ảnh đầu vào một lần và tính toán một bản đồ các đặc trưng. SSD cũng sử dụng các hộp neo ở các tỷ lệ khung hình khác nhau tương tự như Faster R-CNN[18]. Để xử lý tỷ lệ.

SSD dự đoán các hộp giới hạn sau nhiều lần co giãn.

Để dự đoán các hộp giới hạn và xác suất phân loại chúng ta chạy một filter có kích thước nhỏ 3x3 trên toàn bộ bản đồ đặc trưng. Vì mỗi lớp convolutional hoạt động ở một mức độ khác nhau, nó có thể phát hiện các đối tượng ở các tỷ lệ khác nhau.

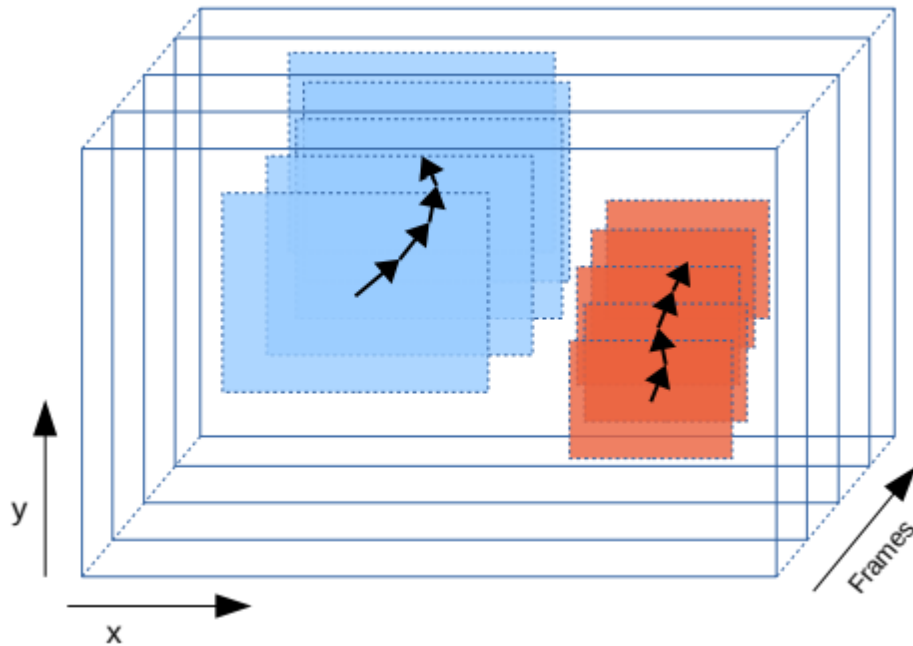
Nhiều phương pháp đánh giá ta có rất nhiều phương pháp để phát hiện phương tiện giao thông. Nhưng do tốc độ di chuyển của các phương tiện giao thông thường cao nên cần một phương pháp có tốc độ chính xác và xử lý nhanh tốt trong điều kiện ánh sáng tốt. Qua những phân tích trên có thể nhận thấy mô hình

**SSD** là một trong những mô hình có hiệu quả tốt. Đối với các vật thể lớn, SSD có thể vượt trội hơn Faster R-CNN và R-FCN về độ chính xác với bộ giải nén nhẹ hơn và nhanh hơn. sử dụng mô hình SSD cân bằng giữa tốc độ xử lý và độ chính xác, khi được áp dụng vào các hệ thống phát hiện phương tiện giao thông phục vụ giám sát và điều chỉnh các hệ thống quản lý giao thông hiện tại.

#### ***2.4.4. Các phương pháp truy vết nhiều vật thể trong video***

##### ***➤ IoUTracker [21]***

Đây là một phương pháp truy vết có nguyên lý đơn giản và đạt được tốc độ xử lý rất cao. IoUTracker mở rộng chuỗi truy vết của một vật thể bất kỳ bằng cách tính mức độ trùng khớp của vị trí xuất hiện cuối cùng của vật đó với toàn bộ các vật thể được phát hiện tại frame hiện tại. Vật có mức độ chồng khớp lớn nhất và lớn hơn một ngưỡng độ tin tưởng sẽ được sử dụng như vị trí kế tiếp cho chuỗi truy vết này. Vị trí của các vật thể tại mỗi frame được biểu diễn bởi các hộp giới hạn hình chữ nhật.



**Hình 10: Nguyên Lý Hoạt Động của IoUTracker**

Một chuỗi truy vết không được nối với bất kỳ vật thể ở frame hiện tại (các giá trị chồng khớp đều nhỏ hơn ngưỡng tin cậy) sẽ được xem như vật thể tương ứng chuỗi này đã rời khỏi khung hình. Một vật ở frame hiện tại không được nối với bất kỳ chuỗi nào trước đó sẽ được xem như một vật vừa đi vào khung hình. Có nhận xét rằng IoUTracker chỉ có 3 trong số 5 bước chính của thuật toán truy vết: phát hiện, tính độ liên quan (ở đây là IOU) và truy vết.

### ➤ ***SORT*** [22]

Phần này sẽ trình bày về SORT, một phương pháp cũng thuộc nhóm Separated Detection and Tracking. So với IoUTracker, SORT có thêm 1 trong số các bước chính của bài toán truy vết là dự đoán chuyển động. Các vật thể đã xuất hiện trong các frame trước đó được dự đoán vị trí mới ở frame hiện tại. Các vật thể vừa mới được phát hiện được so sánh với vị trí dự đoán mới này thay vì là vị trí xuất hiện cuối cùng như trong IoUTracker. Mô hình dự đoán chuyển động được sử dụng là Kalman Filter

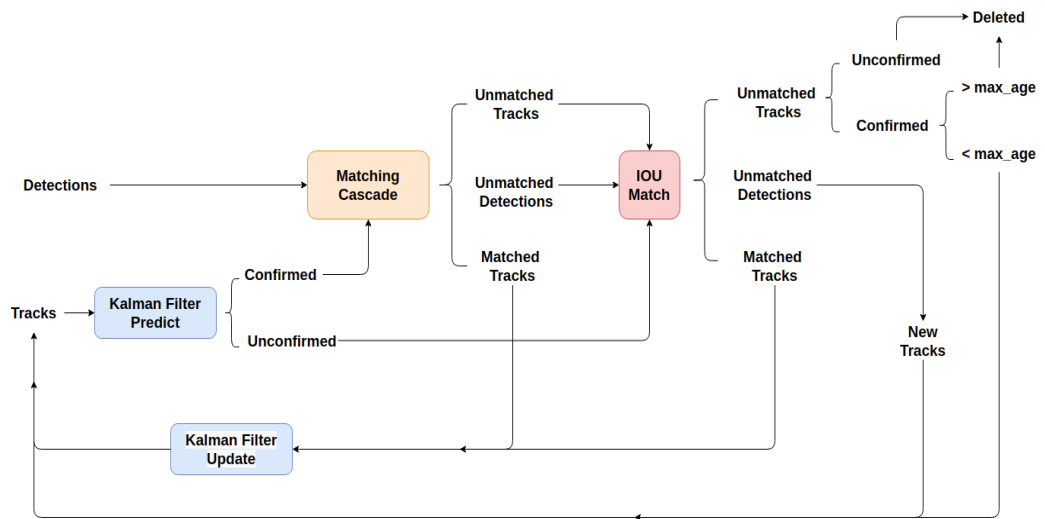
Vấn đề của SORT (và IoUTracker) là chỉ sử dụng thông tin IoU để liên kết các chuỗi với vật thể mới. Hướng thực hiện này chỉ có thể hoạt động hiệu quả khi video có độ phân giải tốt, IoU của các hộp giới hạn thuộc cùng một đối tượng lớn. Các giải pháp như SORT và IoUTracker sẽ kém hiệu quả khi xảy ra tình trạng vật bị che khuất hoặc nhiều vật có quỹ đạo tương đồng..

### ➤ ***DEEPSORT***

Để giải quyết các vấn đề còn tồn tại của SORT, tác giả của bài báo DEEPSORT[23] đã sử dụng thêm các đặc trưng về không gian và đặc trưng trực quan để phân biệt các đối tượng. Ngoài ra, một chiến lược liên kết theo tầng (Matching Cascade) và một cách quản lý mới cho vòng đời của chuỗi truy vết được xây dựng nhằm nâng cao độ chính xác liên kết.

Cải tiến thứ nhất của DEEPSORT là sử dụng thêm các độ đo khác ngoài IoU để liên kết các đối tượng. Các độ đo mới này bao gồm khoảng cách Mahalanobis giữa chuỗi truy vết và vật thể mới phát hiện trong không gian véc-tơ, và khoảng cách Cosine giữa 2 véc-tơ đặc trưng trực quan tương ứng 2 hộp giới hạn. Các đặc trưng trực quan này được trích xuất thông qua mạng học sâu, cụ thể là Wide Residual Net[24] trong bài báo gốc của tác giả.

Cải tiến thứ 2 của DEEPSORT là thay đổi cách quản lý vòng đời chuỗi truy vết. Thay vì loại ngay một chuỗi truy vết khi nó không được kết nối, tác giả của DEEPSORT đề xuất gán cho mỗi chuỗi 1 trong 3 loại trạng thái (tentative, confirmed, deleted):



**Hình 11: Sơ Đồ Phương Pháp DEEPSORT**

Cải tiến thứ 3 của DEEPSORT nằm ở chiến lược liên kết theo tầng. Như đã trình bày ở trên, 1 chuỗi đã được xác nhận sẽ không bị xóa trong  $t_2$  frame. Tuy nhiên, khi liên kết các chuỗi này với kết quả của mô hình phát hiện vật thể trong frame hiện tại, các chuỗi bị mất dấu càng trễ sẽ càng được ưu tiên liên kết trước.

Luồng xử lý của Deep SORT (**Hình 12**) được thực hiện tuần tự qua các bước dưới đây :

Bước 1: Sử dụng Faster Region CNN (với backbone là VGG16) để phát hiện các đối tượng trong khung hình hiện tại.

Bước 2: Deep SORT sử dụng Kalman Filter để dự đoán các trạng thái track mới dựa trên các track trong quá khứ. Các trạng thái này lúc mới khởi tạo sẽ được gán 1 giá trị mang tính thăm dò (tentative). Giá trị này nếu vẫn đảm bảo duy trì được trong 3 frame tiếp theo, trạng thái sẽ chuyển từ thăm dò sang xác nhận (confirmed), và sẽ cố gắng được duy trì theo dõi trong 30 frame tiếp theo. Ngược lại, nếu mất dấu khi chưa đủ 3 frame, trạng thái sẽ bị xóa khỏi trình theo dõi.

Bước 3: Sử dụng những track đã được xác nhận, tiến hành đưa vào chiến lược đối sánh phân tầng (matching cascade) nhằm liên kết với các detection phát hiện được dựa trên độ đo về khoảng cách và đặc trưng.

Bước 4: Các track và các detection chưa được liên kết sẽ được đưa đến 1 lớp lọc tiếp theo. Sử dụng giải thuật Hungary giải bài toán phân công với ma trận sử dụng IOU để liên kết lần 2

Bước 5: Xử lí, phân loại các detection và các track

Bước 6: Sử dụng Kalman filter để hiệu chỉnh lại giá trị của track từ những detection đã được liên kết với track và khởi tạo các track mới.

### ➤ *CenterTrack*

Đây là phương pháp thuộc nhóm Joint Detection and Tracking được tìm hiểu đầu tiên trong phạm vi khóa luận. CenterTrack[24] thực hiện một số thay đổi lên mô hình CenterNet[25] để có thể ứng dụng trực tiếp cho bài toán truy vết đa đối tượng.



**Hình 13: Sơ đồ phương pháp CenterTrack[24]**

Được xây dựng dựa trên CenterNet, CenterTrack dùng thêm ảnh và bản đồ nhiệt từ 1 frame liền trước để bổ sung thông tin cho frame hiện tại. Hai thành phần này cùng với frame hiện tại sẽ tạo thành đầu vào cho mạng. Lập luận cho cách làm này, bài báo cho rằng enterNet tập trung phát hiện vật thể trong 1 ảnh duy nhất. Nếu một vật thể bị che khuất và không thể nhìn thấy ở frame hiện tại, CenterNet sẽ không thể phát hiện được. Bằng cách dùng frame phía trước như đầu vào bổ sung, CenterTrack hướng đến việc phát hiện được các vật nhìn thấy được từ quá khứ nhưng bị khuất ở hiện tại. Đầu ra của mạng ngoại trừ  $Y$  và  $S$  như phiên bản gốc sẽ có thêm một ma trận  $D \in \mathbb{R}^{W/R \times H/R \times 2}$  biểu diễn véc-tơ độ lệch của tâm vật thể so với giá trị từ frame phía trước

Chiến lược liên kết của CenterTrack có thể miêu tả như sau. Mỗi vật thể vừa được phát hiện  $p$  với độ lệch tâm  $d_p$  sẽ được nối với vật thể có khoảng cách gần nhất với  $p - d_p$  nếu khoảng cách này nhỏ hơn ngưỡng  $k$ . Nếu vật  $p$  không nối được với vật nào từ frame trước, một chuỗi truy vết mới ở vị trí của  $p$  sẽ được tạo ra.

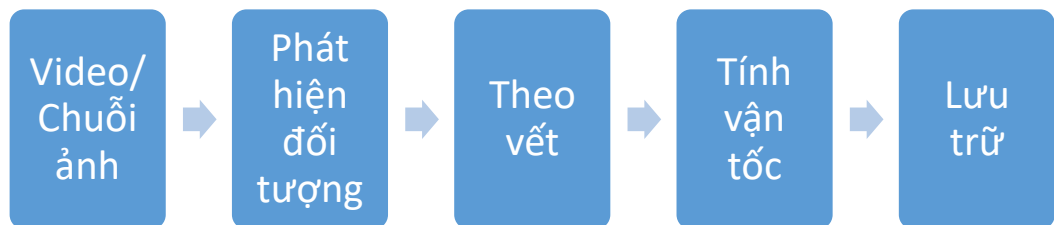
## Chương 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

Trong chương này luận văn sẽ trình bày phát hiện truy vết và đo tốc độ phương tiện giao thông.

### 3.1. Hệ thống phát hiện chuyển động, theo vết phương tiện giao thông và tính tốc độ phương tiện

#### ○ Tổng quan hệ thống

Từ các phân tích tổng quan và các lý thuyết liên quan ở trên, luận văn đề xuất hệ thống phát hiện chuyển động, theo vết phương tiện giao thông và tính tốc độ phương tiện gồm các bước như hình 13:



**Hình 14: Mô hình hệ thống phát hiện chuyển động, theo vết phương tiện giao thông và tính tốc độ phương tiện**

Phát hiện phương tiện giao thông là một hệ thống phức tạp và khó khi cần xác định được tọa độ định vị trí và phân loại nhận biết các phương tiện giao thông. Xây dựng hệ thống phát hiện phương tiện giao thông, theo vết và đo tốc độ phương tiện bao gồm ba bước xử lý chính:

*Bước 1:* Thu nhận dữ liệu đầu vào là các video hình ảnh từ các hệ thống camera giao thông và phân tích thực hiện tiền xử lý dữ liệu đầu vào.

*Bước 2:* Sử dụng một mô hình phát hiện (SSD), nhận dạng phương tiện đã được huấn luyện để phát hiện và trả về kết quả bao gồm hộp giới hạn và đối tượng xuất hiện trong các hộp giới hạn đó.

*Bước 3:* Thực hiện hậu xử lý các video để loại bỏ các hộp chồng chéo, sử dụng phương pháp DEEPSORT theo dõi truy vết các phương tiện và đo tốc độ phương tiện...



*Bước 4:* Lưu trữ hình ảnh các xe tốc độ vượt quá giới hạn.

Phát hiện phương tiện giao thông có thể được áp dụng vào các hệ thống đếm số lượng phương tiện tham gia giao thông nhằm giám sát tình trạng giao thông tại các giao cắt, ngã tư tránh ùn tắc giao thông và các hệ thống xử phạt khi các phương tiện đi sai làn trên các tuyến đường được phân làn riêng cho từng loại phương tiện.

○ ***Các điều kiện ràng buộc***

Để thử nghiệm đạt được kết quả phát hiện chính xác cao các hệ thống camera cần được bố trí tại các vị trí phù hợp như giữa các làn xe để có góc nhìn rộng toàn cảnh trực quan tránh các trường hợp bị che khuất bởi các ngoại vật, các vị trí có ánh sáng tối ưu.

Hệ thống xử lý cần được hỗ trợ các máy tính xử lý có nhiều CPU hoặc GPU giúp cho việc phát hiện tính toán tốc độ các phương tiện nhanh, ít sai số và chính xác đảm bảo thời gian thực.

○ ***Phát triển Hệ Thống Đo Tốc Độ Phương Tiện Giao Thông***

Hệ Thống đo tốc độ phương tiện tham gia giao thông với mục đích là phân tích xử lý nhanh nhận diện và tính toán được tốc độ phương tiện hệ thống đo tốc độ phương tiện tham gia giao thông có 2 thách thức lớn:

- Khả năng hoạt động tự động, nói cách khác là chỉ cần lắp đặt, hệ thống sẽ tự động căn chỉnh camera sao cho tính toán được tốc độ phương tiện một cách chính xác bởi nếu với mỗi camera đều phải cung cấp hoặc đo đạc các tham số thủ công thì sẽ rất tốn chi phí cả về nhân lực và vật lực;

- Khả năng hoạt động ở các góc nhìn khác nhau để tận dụng được tối đa các máy quay giám sát đã được lắp đặt.

Về các bước xử lý, Hệ thống đo tốc độ phương tiện tham gia giao thông nhìn chung sẽ có các bước xử lý gồm đầu tiên là xác định phương tiện, sau đó sử dụng các kỹ thuật để tìm kiếm đoạn đường di chuyển của phương tiện, cuối cùng tốc độ sẽ được tính bằng khoảng cách di chuyển chia cho thời gian. Về vấn đề thời gian ta có thể dễ dàng tính toán được khi nào đầu vào của hệ thống là các video.

Tuy nhiên việc tính toán khoảng cách đoạn đường di chuyển lại là một vấn đề khó khăn do khoảng cách trong video 2 chiều và khoảng cách trong không gian thực 3 chiều là khác nhau vì vậy cần phải có một bước căn chỉnh để khớp tọa độ của camera giám sát, tọa độ thực và tọa độ trên ảnh.

### **3.2. Phương pháp phát hiện chuyển động, theo vết phương tiện giao thông và tính tốc độ phương tiện**

#### ***- Phương pháp xác định đối tượng đối tượng và theo vết đối tượng***

Thuật toán SSD là một thuật toán khá phức tạp, có nhiều tầng và các giai đoạn xử lý khác nhau. Sau khi tìm hiểu và phân tích thuật toán, tôi tiến hành thực hiện xây dựng mô hình nhận diện phương tiện giao thông.

Bộ SSD (SINGLE SHOT DETECTOR) dựa vào sự tương ứng của mỗi đối tượng từ khung hình này sang khung hình khác để xác định vị trí của đối tượng trong khung hình tiếp theo

Với mục đích xây dựng mô hình nhận diện phương tiện giao thông nhằm mục đích làm nền tảng cho việc xây dựng ứng dụng. Mô hình yêu cầu các đặc tính sau:

- Nhận diện được các phương tiện giao thông qua hình ảnh, video.
- Nhận diện được các phương tiện ở trong thời gian thực.

Mô hình đảm bảo tốc độ xử lý nhận diện nhanh, ổn định và đạt độ chính xác 80%

Từ những phương pháp theo vết đối tượng được trình bày ở mục 2.4.4 chương trình áp dụng phương pháp DEEPSORT để theo vết đối tượng

Với những yêu cầu chung từ mô hình, mô hình nhận diện các phương tiện giao thông bằng thuật toán Single Shot Multibox Detector (SSD) và sử dụng phương pháp DEEPSORT để theo vết đối tượng phải đảm bảo các tính năng sau:

- Xử lý được hình đầu vào.
- Hiện thị được đặc trưng các phương tiện nhận diện được
- Theo vết nhận diện theo từng khung hình

Với những yêu cầu của hệ thống đã đề ra, tiến hành đưa ra các phương án xử lý như sau:

- Chuẩn bị bộ dữ liệu để nhận dạng đối tượng.
- Xây dựng quá trình tiền xử lý ảnh đầu vào.
- Dò tìm vùng chứa các đối tượng.
- Nhận diện các đối tượng.
- Truy vết đối tượng.
- Kết quả.

- ***Phương pháp luồng quang xác định vận tốc (optical flow)***

Optical flow hay còn gọi là luồng quang lượng là một kỹ thuật trong thị giác máy tính

Phương pháp optical flow được vận dụng thực hiện bằng cách sử dụng các tập hợp vector có hướng của các đối tượng chuyển động theo thời gian để phát hiện các vùng chuyển động trong một ảnh.

Luồng quang (optical flow): dùng để nhận biết sự chuyển động tương đối của các điểm ảnh trên bề mặt nền ảnh của đối tượng dưới góc nhìn từ một vị trí cố định. Từ một tập điểm cho trước, tìm một tập điểm tương tự trên một ảnh khác dựa trên sự thay đổi không lớn về cường độ sáng (intensity) của điểm ảnh, optical flow xác định vùng chuyển động trong ảnh và ước lượng sự thay đổi vị trí, vận tốc của điểm ảnh theo thời gian.

- ***Xác định vận tốc dựa vào phương pháp khung đo***

Phương pháp khung đo là một vùng hình chữ nhật được xác định bằng 4 điểm đánh dấu trước trên làn đường, dùng để xác định vị trí, kích thước của vùng cần giám sát. Từ những điểm đã được đánh dấu tạo thành 2 vạch gồm vạch phát hiện đối tượng đi vào vùng tạo sẵn và vạch phát hiện đối tượng đi ra khỏi vùng đo.

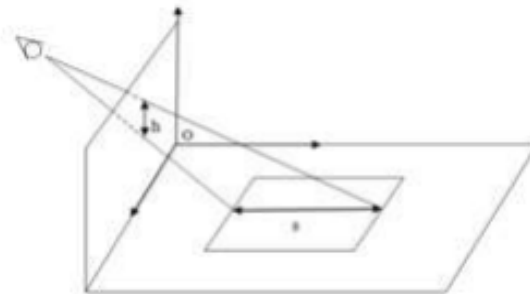
Chiều dài khoảng cách của 2 vạch tương ứng với chiều dài thực tế và phương hướng của đoạn đường cần quan sát. Khi đối tượng đi vào vùng vạch đầu tiên, hệ

thống sẽ tự tính toán ghi nhận vị trí di chuyển của đối tượng qua từ frame ảnh làm cơ sở theo dõi chuyển động của đối tượng.



**Hình 15: Mô hình khung đo**

Các tập hợp điểm  $x_1, x_2, x_3, x_4$  (Hình 14:) là các điểm mốc được vẽ sẵn để dễ quan sát trên làn đường thực tế,  $s$  là chiều dài (mét) của đoạn đường được xác định.



**Hình 16: Tỷ lệ chiều dài thực và chiều dài khung đo [3]**

Vận tốc được tính qua camera chủ yếu dựa trên sự thay đổi vị trí của từng đối tượng trên khung đo, mỗi khoảng di chuyển trong khung đo tương ứng trên màn hình máy tính được tính bằng các pixel, vì vậy cần khoảng thời gian quá trình ánh xạ chiều dài các pixel thành chiều dài thực tế thông qua tỷ lệ giữa khung hình, chiều dài  $h$  và chiều dài thực tế  $s$  của khung đo như hình 15:

$$C = \frac{s}{h} \quad (14)$$

c: tỷ lệ chiều dài thực tế và chiều dài của khung đo

h: chiều dài khung đo được tính (pixel)

s: chiều dài thực được tính (mét)

Đối với các đối tượng di chuyển thời gian được tính dựa trên số khung hình được phát trong 1 giây đối với xử lý trên khung hình video của đối tượng.

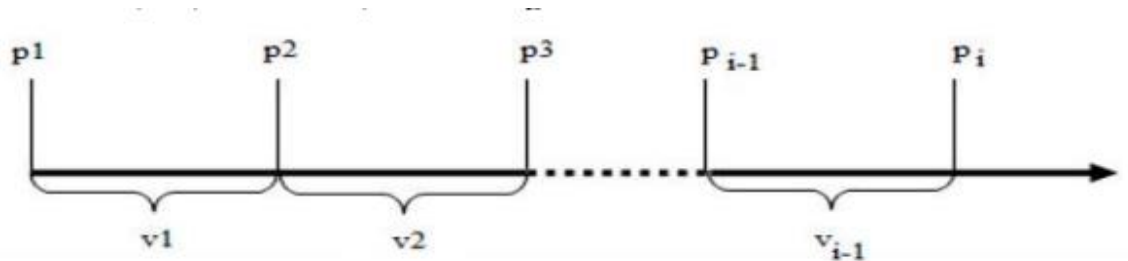
$$\text{Video: } \nabla t = \frac{1}{fps} \quad (15)$$

$\nabla t$ : thời gian được tính chuyển đổi giữa 2 frame

fps: số frame tính trên giây

Khi xe bắt đầu đi vào và ra khỏi khung hình đo phải trải qua nhiều frame, mỗi frame sẽ ghi lại một vị trí tùy thời gian thực của vật.

Khi áp dụng tính vận tốc trung bình dựa trên việc tính vận tốc của từng đối tượng khi di chuyển qua từng cặp frame liên tiếp nhau.



**Hình 17: Tính vận tốc trung bình của các giai đoạn [1]**

Khoảng cách mà đối tượng di chuyển trên 2 frame liên tiếp theo hình 16: dựa trên khoảng cách của điểm tính  $p_{i-1}(x_1, y_1)$  của đối tượng ở frame trước và đối tượng điểm  $p_i(x_2, y_2)$  ở frame sau, với  $d$  là khoảng cách của đối tượng  $p_{i-1}, p_i$ , ta có thể tính được khoảng cách thực tế mà đối tượng đi được

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (16)$$

$$s = d * c \quad (17)$$

$$v = \frac{s}{t} \quad (18)$$

$$v_{avg} = \frac{\sum_{i=1}^n v_i}{n} \quad (19)$$

$v_{avg}$ : vận tốc trung bình của đối tượng

$v_i$ : vận tốc thực tế của đối tượng di chuyển từ frame  $i-1$  đến frame  $i$

$d$ : khoảng cách giữa hai điểm  $p_{i-1}$ ,  $p_i$

$s$ : khoảng cách đã đi trong thực tế của đối tượng

$c$ : tỷ lệ chiều dài

$n$ : số lược vận tốc đo được trên khung thực tế

Như vậy từ hai phương pháp đã nêu trên luận văn chọn phương pháp xác định vận tốc dựa vào phương pháp khung đo vì phương pháp này có khả năng hoạt động tự động, các góc camera có thể tự căn chỉnh góc độ phù hợp với môi trường hoạt động, có khả năng hoạt động linh hoạt ở nhiều môi trường khác nhau, góc quay thông dụng với nhiều máy quay giám sát đã được lắp đặt, quan trọng nhất là các thông số kỹ thuật đã có sẵn việc cấu hình cài đặt áp dụng dễ dàng, khó khăn của phương pháp này là phải quan sát phương tiện để hoàn thành việc hiệu chuẩn.

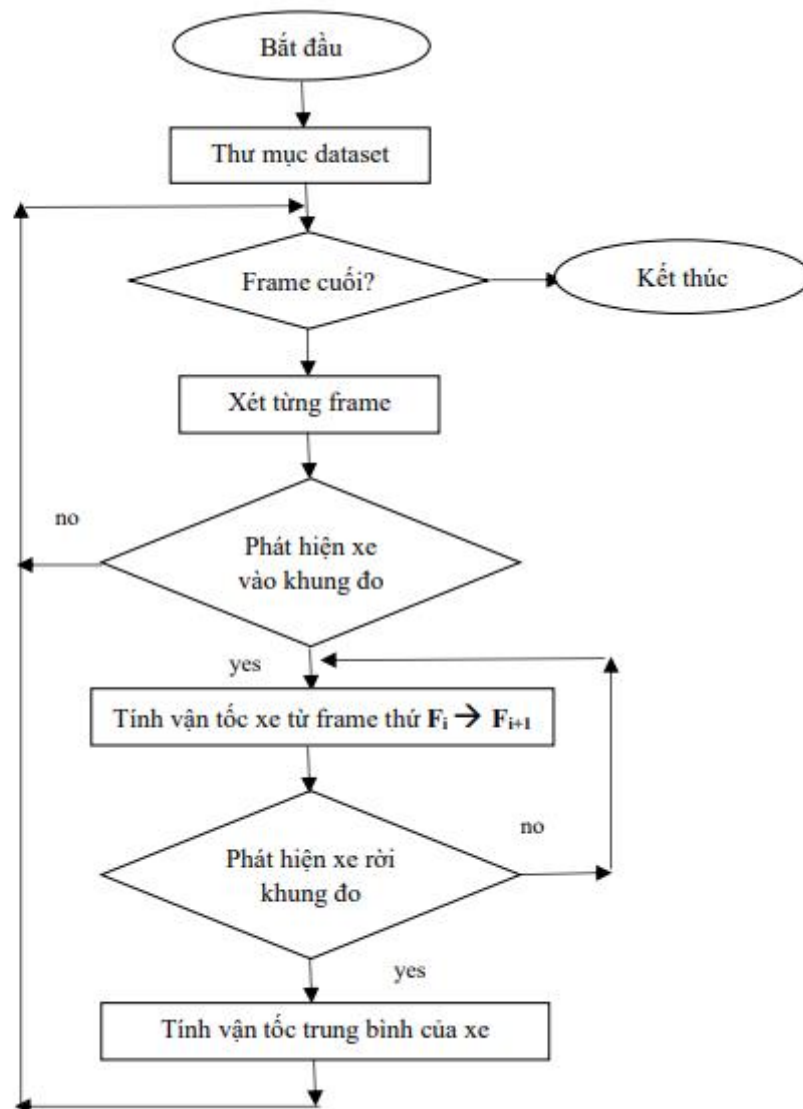
### 3.3. Xây dựng chương trình

#### - Sơ đồ khối

Lưu đồ hoạt động của hệ thống đo tốc độ phương tiện tham gia giao thông sử dụng camera hình ảnh được đề xuất trong luận văn như hình 17: bao gồm

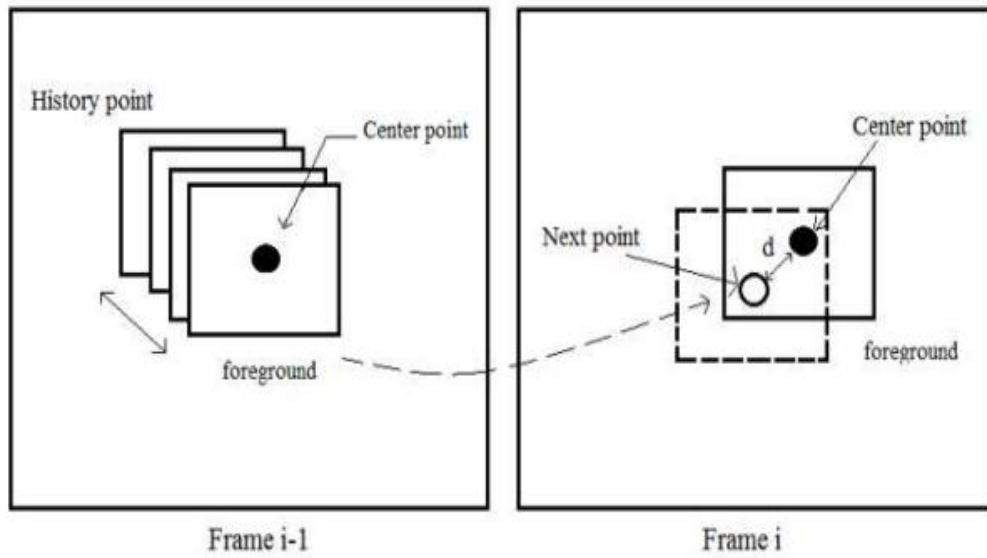
Input: dữ liệu từ video.

Output: vận tốc của các đối tượng.



**Hình 18: Lưu đồ hoạt động**

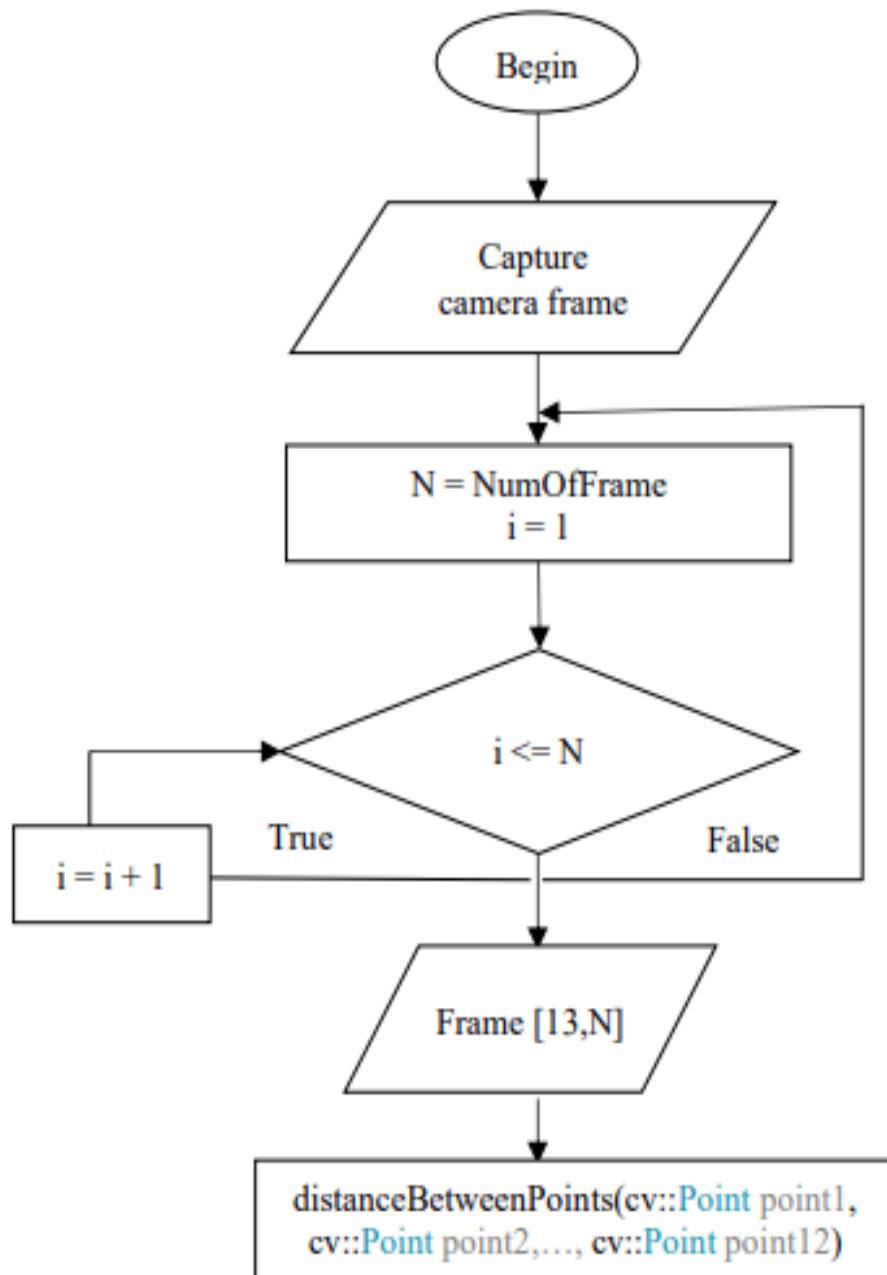
Mỗi frame trong video là một hình ảnh độc lập, cùng một đối tượng xuất hiện ở nhiều frame sẽ có những vị trí khác nhau, vì vậy cần xác định đối tượng này là duy nhất trên tất cả các frame dựa vào sự thay đổi không quá lớn về khoảng cách của đối tượng này trong 2 frame liên tiếp minh họa nhận dạng xác định đối tượng qua từng frame như hình 18:



**Hình 19: Minh họa xác định đối tượng**

Trong quá trình từ khi phương tiện bắt đầu đi vào và đi ra khỏi khung đo trải qua nhiều frame, mỗi frame ghi lại một vị trí. Việc tính vận tốc trung bình dựa trên việc tính vận tốc của đối tượng khi di chuyển qua từng cặp frame liên tiếp sau đây là lưu đồ tính khoảng cách di chuyển thực tế của hai đối tượng trên các frame như hình 19:





Hình 20 : Lưu đồ thuật toán tính khoảng cách di chuyển thực tế của hai đối tượng trên các frame

### 3.4. Công cụ thực hiện

Sử dụng công cụ lập trình Python 3.7 và thư viện OpenCV, Tensorflow

Chương trình demo được viết bằng ngôn ngữ Python. Dựa trên yêu cầu thực tế để dự đoán tốc độ xe chuyển động bằng camera thì chương trình cần phải có các chức năng như: lấy dữ liệu đầu vào là các tập tin video có sẵn từ các camera giao thông trên các quốc lộ, cửa ngõ ra vào các trục đường chính, dựa vào tập tin video được chọn và vị trí góc đặt của camera để điều chỉnh các thông số bao gồm: ngưỡng min, ngưỡng max, khung hình... sau khi thiết lập các thông số sẽ bắt đầu chạy chương trình demo để hiển thị đoạn video ra khung hình chính và bật đo tốc độ sau khi vẽ khung hình trong đoạn video và xuất hình ảnh xe vượt quá giới hạn tốc độ, tốc độ được hiển thị trên giao diện chương trình.

Giao diện chương trình như hình 20 bao gồm:

1. Khung hình chính hiển thị video trích xuất từ camera.
2. Hình ảnh xe có tốc độ vượt giới hạn.
3. Tốc độ đo được của xe.



**Hình 21: Giao diện chương trình chạy video**

Phần giao diện chương trình chạy video ta thấy các line red, green, blue, bounding box có tác dụng gì?

- Green line dùng để xác định xe vào
- Blue line dùng để xác định xe ra
- Red line dùng để xác định biên số để chụp lại.
- bounding box dùng để nhận dạng theo vết khi đối tượng chuyển động

Từ giao diện chương trình (**Hình 22**) cách xác định tốc độ dựa trên các Green line, Blue line. Tỷ lệ của các line dựa trên chiều cao và góc độ của máy quay được căn chỉnh và so sánh với thiết bị đo tốc độ chuyên dụng.

Ta tính được vận tốc dựa trên các frame của đối tượng từ lúc đi vào từ Green line và lúc kết thúc Blue line, tốc độ dựa trên tốc độ ghi hình từng frame của thiết bị do chia cho thời gian của frame Blue line – frame Green line ta được tốc độ của phương tiện.

$$v = \frac{\text{fps}}{(\text{Blue Time} - \text{Green Time})} \quad (20)$$

v: Tốc độ phương tiện

fps: Tốc độ ghi của thiết bị fps/giây

BlueTime: Thời gian đi ra của đối tượng

GreenTime: Thời gian đi vào của đối tượng

## **Chương 4: THỬ NGHIỆM VÀ ĐÁNH GIÁ**

Thực hiện xây dựng bộ dữ liệu từ các camera giao thông tại thành phố Tây Ninh. Tiến hành thực hiện huấn luyện tập dữ liệu với hệ thống tự xây dựng, đánh giá kết quả và thực hiện hiệu chỉnh, cải tiến nếu có.

### **4.1. Thu thập và tiền xử lý dữ liệu**

Bộ dữ liệu thử nghiệm được thu thập từ video giao thông được cung cấp bởi hệ thống camera giao thông của thành phố Tây Ninh.

Dữ liệu thu thập từ máy chủ VNPT tây ninh, 1 camera khu vực cổng chào thành phố Tây Ninh, bộ dữ liệu chạy thử nghiệm 140 GB, với tổng số giờ là 120 giờ, có tốc độ ghi 25 fps/giây, kích thước video là 1920x1080 với định dạng file .mp4. Trong bộ dữ liệu này có một khoảng thời gian được học viên mượn máy bắn tốc độ để đo thực tế và để so sánh với kết quả của chương trình thử nghiệm và sẽ trình bày ở phần sau.

### **4.2. Cài đặt và thử nghiệm**

#### **4.2.1. Thiết lập cấu hình**

Chương trình được xây dựng trên hệ thống với các thông số như sau:

✓ Bộ xử lý Intel Core i9-9900k, 3.60GHz, 32GB RAM, GPU 1050 TI

✓ Hệ điều hành Windows 10 64bit.

Google Colab Pro GPU Tesla P1 với bộ nhớ 16280MiB

File video được lấy từ máy chủ camera giám sát giao thông tây ninh của VNPT tây ninh.

- Camera quay quét (HIKVISION) thực tế như hình 21: đặt tại điểm quốc lộ 22B (thành phố tây ninh)



**Hình 23: Camera quay quét**

#### ***4.2.2. Dữ liệu thực nghiệm***

Trong luận văn sử dụng dữ liệu 240 GB tập tin video để thử nghiệm, dữ liệu được trích xuất từ camera cửa ngõ quốc lộ vào thành phố tây ninh. Dữ liệu được lấy từ máy chủ VNPT tây ninh

#### ***4.2.3. Quá trình và kết quả thực nghiệm***

##### ***➤ Quá trình thực nghiệm***

Địa điểm thực nghiệm tại vị trí có gắn camera giám sát giao thông, camera ghi hình được đặt ở độ cao 6m, camera được đặt trên cao vuông góc giữa các làn đường, với góc quay 30° độ so với mặt đường, có tốc độ ghi 25fps/giây, kích thước video là 1920x1080 với định dạng file .mp4. Thực hiện bằng cách sử dụng máy đo tốc độ chuyên dụng và camera đặt ở vị trí như trên để ghi hình lại tất cả các xe chạy qua sau đó dùng chương trình demo sử dụng các file đã ghi hình để thử nghiệm sau đó đối chiếu kết quả so với máy đo tốc độ bao gồm các phần như sau:

- Phát hiện đối tượng chuyển động: cho thấy hình khoanh vùng các đối tượng chuyển động.
- Theo vết đối tượng: Mỗi xe được gán 01 ID và theo được vết của xe.
- Đo tốc độ: Có thể đo tốc độ và hiển thị kết quả lên chương trình demo.
- Xuất kết quả từ chương trình kết hợp với tốc độ từ máy đo so sánh đối chiếu

### ➤ *Chuẩn bị dữ liệu*

Để huấn luyện mô hình, dataset cần được gán nhãn, tuy nhiên dữ liệu thu được từ VNPT Tây Ninh lại chưa được gán nhãn. Do đó để thuận tiện và đỡ mất thời gian gán nhãn, học viên đã khảo sát và nhận thấy bộ dataset KITTI là một bộ dữ liệu có gán nhãn, có chất lượng tốt và đã được cộng đồng thế giới dùng làm thước đo các giải thuật máy học trong việc phát hiện và theo vết đối tượng. Từ đó Dataset KITTI được lựa chọn để huấn luyện và đánh giá mô hình. Ngoài ra trong luận văn sử dụng SSD và bộ dữ liệu Pascal VOC được dùng để huấn luyện mạng gốc SSD đã bao gồm các phương tiện giao thông, nên việc đào tạo tương đối khá nhanh chóng.

KITTI là bộ dữ liệu về giao thông đường bộ được xây dựng cho việc huấn luyện trong máy học và thị giác máy tính.

Trong bộ dữ liệu KITTI gồm có 7481 hình ảnh huấn luyện, 7518 hình ảnh kiểm tra và 80256 đối tượng được dán nhãn. Tôi đã chia bộ dữ liệu huấn luyện gồm 7500 hình ảnh trong đó dùng 10% để kiểm thử và 90% để huấn luyện.

### ➤ *Xây dựng model*

Model được tìm hiểu và xây dựng từ mã nguồn mở, nguồn được tham khảo từ github[26][27]

Dữ liệu được huấn luyện với Google Colaboratory Pro trên mô hình SSD tập dữ liệu train được trích xuất từ bộ dữ liệu Dataset KITTI. Bộ dữ liệu thử nghiệm chạy với tổng thời gian trên 35 phút.

Gán nhãn dữ liệu bằng phần mềm LblImage được lưu dưới dạng tệp XML ở định dạng PASCAL VOC.

Sau khi chọn được mã nguồn Model cần huấn luyện, ta tiến hành chọn tài nguyên và đăng ký Colab Pro US\$ 9,99/tháng

Chọn tải toàn bộ dữ liệu XML và dữ liệu cần huấn luyện lên Google Drive sau đó ta tiến hành kết nối và huấn luyện dữ liệu với mô hình SSD

Phân chia tập dữ liệu huấn luyện và kiểm tra, chuyển định dạng từ XML sang CSV tải các Detection API, tải mô hình SSD và tiến hành cấu hình cài đặt thông số cho mô hình

Bước tiếp theo huấn luyện mô hình, xuất kết quả và kiểm tra mô hình.

Bộ dữ liệu huấn luyện được thử nghiệm nhiều lần để tìm ra độ chính xác phù hợp.

- huấn luyện 1000 bước thì ta thấy sai số còn cao: 0.3759675
- huấn luyện 10000 bước thì sai số gần như đã thấp và có thể dùng được: 0.16466631

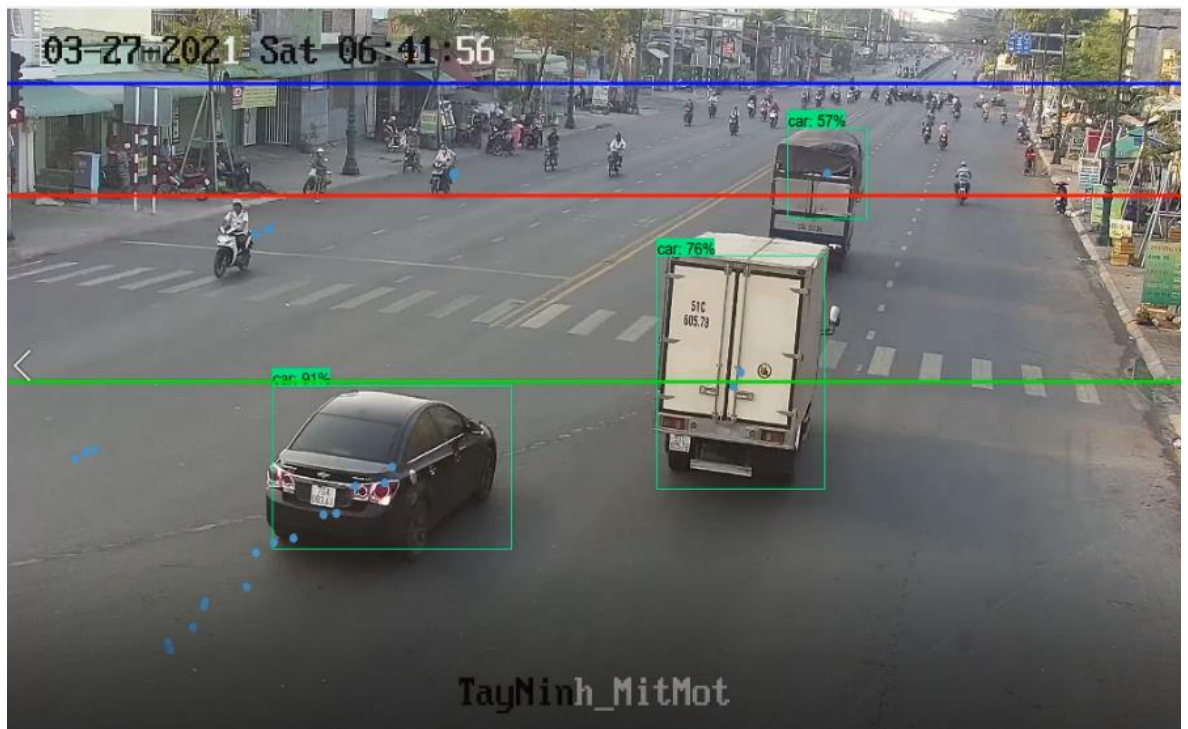
Sau khi huấn luyện so sánh các bước thì ta nhận thấy độ sai số 0.16466631 của tập huấn luyện 10000 bước có thể ứng dụng tốt cho chương trình.

Chạy thử nghiệm trên bộ dữ liệu thực tế ở Tây Ninh, trích xuất video thành phố thông minh do VNPT phát triển. Bộ dữ liệu có tổng dung lượng 240GB, có tốc độ ghi 25fps/giây, kích thước video là 1920x1080 với định dạng file .mp4

### ***Kết quả thử nghiệm***

Trong luận văn kết quả thử nghiệm sẽ được trình bày theo 02 loại thử nghiệm:

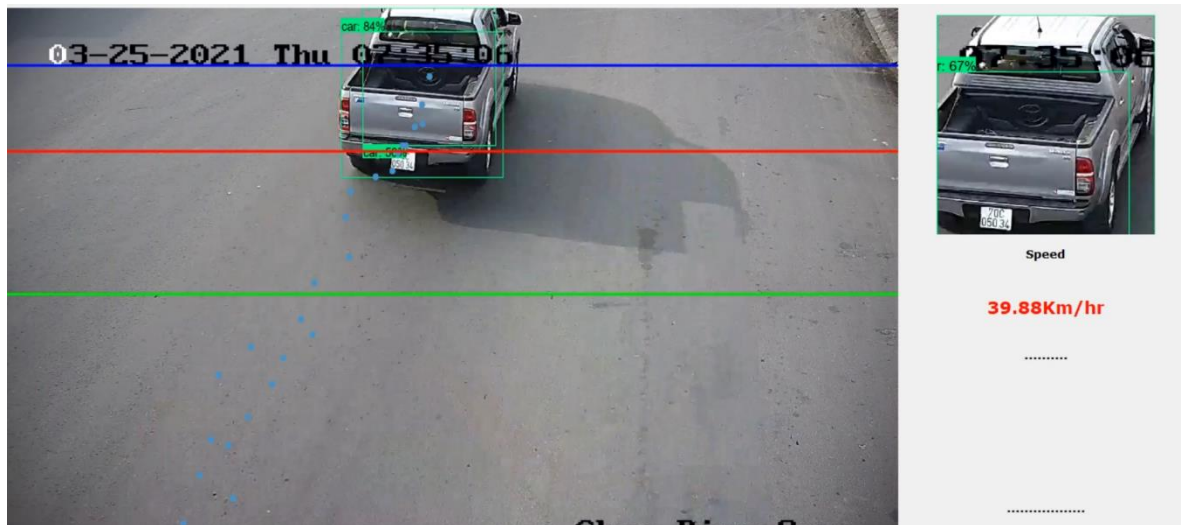
- Thử nghiệm phát hiện đối tượng chuyển động và theo vết đối tượng.



**Hình 24: Phát hiện và theo vết đối tượng chuyển động**

Khi chạy chương trình thì những đối tượng đang chuyển động trong khoản frame như hình 22 thì chương trình phát hiện được đối tượng chuyển động. Các đối tượng sẽ được vẽ một hình vuông – chữ nhật bao quanh có màu xanh và khi theo vết đối tượng này sẽ được đánh số định danh, được nhận dạng theo loại xe và có biểu diễn những chấm xanh khi đối tượng di chuyển qua từng frame.

Thử nghiệm đo tốc độ và hiển thị kết quả lên chương trình.



**Hình 25: Kết quả thử nghiệm đo tốc độ**

Trong phần thử nghiệm như hình 23 ta thấy được sau khi các đối tượng được vẽ bởi hình Vuông – hình chữ nhật bao quanh màu xanh nằm trong các đường ranh giới hạn, khi mà xe chạy quá tốc độ giới hạn khi qua đường màu xanh thì chương trình tự động lưu hình ảnh xe lại và hiển thị kết quả bên phải chương trình và lưu hình ảnh vào hệ thống.

Trong file thử nghiệm này chưa được tích hợp chức năng nhận dạng bản số và xuất execl phương tiện vi phạm.

Video trích xuất để chạy demo có hơn 30 ô tô, thời gian chạy thử nghiệm trên video khoảng 35 phút, lý do khoản thời gian này có dùng máy đo tốc độ chuyên dụng. Có 1 số phương tiện bị bỏ qua hoặc không lấy được tốc độ do có một số phương tiện quá khổ thời gian vào và ra video không được chính xác.

Không có trường hợp theo vết nhầm đối tượng này với đối tượng khác



Có ít phương tiện phát hiện truy vết nhưng không lấy được tốc độ lý do phương tiện quá khổ, đi không đúng làn đường,....

**Bảng 2: Kết quả so sánh vận tốc từ camera bắn tốc độ thật của CSGT và vận tốc đo được của chương trình**

STT	BIỂN SỐ	Loại xe	Tốc độ đo được bằng máy bắn tốc độ của CSGTV (km/h)	Tốc độ đo được trên hệ thống thử nghiệm V' (km/h)	Sai số tốc độ đo $( V-V' /v)*100\%$
1	021.58	XE ÔTÔ	50	44	12 %
2	001.43	XE ÔTÔ	31	26	5
3	041.70	XE ÔTÔ	22	22	8
4	1926	XE ÔTÔ	33	30	3
5	204.56	XE ÔTÔ	38	41	3
6	130.20	XE ÔTÔ	52	56	4
7	2006	XE ÔTÔ	49	33	6
8	105.34	XE ÔTÔ	46	42	4
9	159.74	XE ÔTÔ	40	35	5
10	160.36	XE ÔTÔ	49	51	2
11	606.11	XE ÔTÔ	41	36	5
12	0282	XE ÔTÔ	35	37	2
13	059.61	XE ÔTÔ	39	42	3
14	018.93	XE ÔTÔ	54	50	4
15	175.55	XE ÔTÔ	34	30	4
16	071.39	XE ÔTÔ	35	30	5
17	354.08	XE ÔTÔ	48	40	8
18	000.13	XE ÔTÔ	55	58	3

19	255.69	XE ÔTÔ	39	32	7
20	165.87	XE ÔTÔ	40	35	5
21	732.11	XE ÔTÔ	40	37	3
22	050.34	XE ÔTÔ	45	41	4
23	013.22	XE ÔTÔ	54	51	3
24	017.17	XE ÔTÔ	45	41	4
25	2322	XE ÔTÔ	36	32	4
26	013.98	XE ÔTÔ	38	41	3
27	004.80	XE ÔTÔ	45	48	3
28	112.87	XE ÔTÔ	32	36	4
29	232.11	XE ÔTÔ	35	40	5
30	175.01	XE ÔTÔ	40	36	4
<b>Trung bình</b>					<b>4,3</b>

Từ kết quả thử nghiệm ở trên cho thấy:

- Trên 1 khung hình có trên 2 phương tiện thì hệ thống vẫn phát hiện và theo vết được các phương tiện, không có trường hợp lấy đối tượng này gán cho đối tượng khác.

- Khi có nhiều phương tiện trên khung hình bị phương tiện khác che khuất chương trình vẫn phát hiện đối tượng tuy nhiên có ít trường hợp không tính được tốc độ của phương tiện

- Sau khi chạy demo chạy video trên 30 ô tô đáp ứng các yêu cầu thì có khoản 2 ô tô không nhận được kết quả đo tốc độ tỷ lệ sai số 6,6%

Phương pháp thực hiện đo tốc độ phương tiện chuyển động bằng camera có thể tận dụng các camera có sẵn và đo được trên nhiều làn đường so với camera bắn tốc độ truyền

Tuy nhiên khi sử dụng chương trình chạy thử để đo vận tốc thì cho kết quả đo so với thực tế chính xác khoảng 89.1%, theo tôi thì có những nguyên nhân sau:

Các vị trí camera được đặt góc độ khác nhau do đó dẫn đến việc dự đoán vị trí và lưu vết di chuyển của đối tượng sẽ không chính xác.

Những đối tượng quá lớn thì chương trình chưa nhận dạng không chính xác dẫn đến vận tốc đo chưa đúng.

Việc tính toán vận tốc chỉ đúng cho chuyển động thẳng trong gian 2D (chiều  $x,y$ ) không chính xác khi đối tượng chuyển động theo quỹ đạo khác.

Từ nhiều lần chạy thử nhiệm kết quả trên cho độ chính xác khoảng 89.1% với kết quả này có thể ứng dụng luận văn vào thực tiễn cho đơn vị hoặc địa phương tuy nhiên để có thể ứng dụng vào thực tiễn cần phải cải tiến thêm về nhiều mặt như: thuật toán trừ nền nâng cao để tăng độ chính xác tính toán, có thể nâng chất lượng camera, ứng dụng thuật toán để tính toán vận tốc đo tốc độ thông qua camera với thời gian thực để nhắc nhở xử phạt trực tiếp khi phương tiện vi phạm giao thông.

### **4.3. Kết luận chương**

Trong chương này đã trình bày về quá trình tiến hành thử nghiệm và đánh giá kết quả của hệ thống đề xuất. Các kết quả đạt được khá tốt và có thể phát triển tiếp để ứng dụng vào thực tế tại tỉnh Tây Ninh cũng như mở rộng ra các vùng khác.

Xây dựng nghiên cứu phát triển hệ thống đo tốc độ trên quốc lộ thuộc địa bàn tỉnh Tây Ninh.

Trong đề tài này, tôi đã tìm hiểu ứng dụng các phương pháp nhận dạng vật thể đơn giản có hiệu quả cao, và các phương pháp truy vết tiên tiến cho bài toán ước tính tốc độ phương tiện giao thông. Đồng thời, chúng tôi xây dựng demo cho kết quả của bài toán ước tính tốc độ.

Trong điều kiện dữ liệu có số FPS cao, ít bị che lấp và mô hình phát hiện vật thể SSD được huấn luyện phù hợp, các kỹ thuật đơn giản có tốc độ thực thi cao như DEEPSORT vẫn đạt được độ chính xác truy vết tốt.

Việc sử dụng các kỹ thuật truy vết mới có thể cải tiến đáng kể kết quả phát hiện và theo dõi. Các phương tiện được phát hiện một cách đầy đủ và chính xác hơn.

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Về mặt lý thuyết, luận văn đã hoàn thành được việc tìm hiểu các lý thuyết phục vụ cho hệ thống đo tốc độ xe ô tô chuyển động bằng camera như: Tìm hiểu, nghiên cứu và sử dụng thư viện OpenCV, Tensorflow, Google Colab, hiểu được các khái niệm cơ bản về xử lý ảnh, các thuật toán theo vết đối tượng, nhận dạng đối tượng, chuyển động bằng phương pháp trừ nền và cách thức xác định vận tốc ô tô chuyển động.

Về thực tiễn áp dụng, luận văn đã hoàn thành được mục tiêu được đề ra là: Xây dựng hệ thống giám sát giao thông dựa trên hệ thống giám sát đô thị thông minh tỉnh tây ninh, từ các camera hệ thống Phát hiện, nhận dạng, truy vết đối tượng, xác định được vận tốc đối tượng chuyển động, hoàn thành cài đặt và chạy thử thành công các thuật toán đã nghiên cứu trong phần lý thuyết.

Từ các kết quả chạy thử nghiệm thực tế trên địa bàn tỉnh Tây Ninh ta có thể áp dụng thực tế hệ thống giám sát phát hiện truy vết và cảnh báo tốc độ của phương tiện tham gia giao thông trên địa bàn tỉnh Tây Ninh

Tuy nhiên còn nhiều hạn chế chưa thể áp dụng như là: số lượng camera đáp ứng cho hệ thống còn thiếu, chất lượng của camera chưa đạt yêu cầu, về phần cứng chưa đáp ứng được khi chương trình xử lý cùng lúc nhiều hình ảnh có độ phân giải lớn, chương trình được xây dựng chỉ là phần demo của các thuật toán xác định vận tốc của đối tượng chuyển động.

### Hướng phát triển

- Xác định tốc độ phương tiện với thời gian thực
- Tích hợp thêm nhận dạng bản số
- Tối ưu hóa và loại bỏ các thuộc tính nhiễu
- Tích hợp vào hệ thống giám sát thông minh của tỉnh
- Tìm hiểu thêm thuật toán xử lý ảnh và các phương pháp trừ nền tối ưu hóa chương trình để tính được tốc độ chính xác và nhanh hơn

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Lâm Hữu Tuấn, Huỳnh Phụng Toàn, Trần Cao Đệ, Nguyễn Thị Hồng Nhung, “Dự đoán hướng di chuyển và xác định tốc độ xe qua camera quan sát”, Khoa Công nghệ thông tin & Truyền thông, Đại học Cần Thơ, 2016.
- [2] Nguyễn Văn Trung “Xác định vận tốc đối tượng chuyển động qua camera”, tóm tắt luận văn thạc sĩ, Đại học Đà Nẵng, 2012.
- [3] Viet-Hoa Do, Le-Hoa Nghiem, Ngoc Pham Thi, Nam Pham Ngoc ”A *simple camera calibration method for vehicle velocity estimation*” 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)
- [4] Vũ Đức Lung, Trần Hoàng Lộc, Nguyễn Khắc Ngọc Khôi, Phan Đình Duy “*Khảo sát bài toán nhận diện phương tiện và đo tốc độ phương tiện tham gia giao thông*” tại hội nghị FAIR 2020
- [5] Tài Liệu về Xử Lý Ảnh của PGS TS Đỗ Năng Toàn Và TS Phạm Việt Bình của trường Đại Học Thái Nguyên biên soạn.
- [6] Nguyễn Thị Lan Hương “*Phát hiện, phân loại, theo dõi đối tượng chuyển động trong hệ thống giám sát thông minh*”, khóa luận tốt nghiệp đại học, Đại học Công nghệ - Đại học Quốc gia Hà Nội, 2006.
- [7] Ngô Đức Vĩnh “*Kỹ thuật xử lý vùng quan sát và phát hiện bất thường của các đối tượng trong hệ thống camera giám sát*”, luận án tiến sĩ, Viện Công nghệ thông tin - Viện Hàn lâm Khoa học và Công nghệ Việt Nam, 2016.
- [8]. Jakub Sochor, Roman Juranek, Jakub Spa nhel Luk as marsik,Adam Siroky Adam Herout, Pavel ZemCik in *2012 IEEE 75th Vehicular Technology Conference (VTC Spring)*
- [9]. Dominik Zapletal, Adam Herout “Vehicle Re-Identification for Automatic Video Traffic Surveillance”. In: *International Workshop on Automatic Traffic Surveillance (CVPR 2016)*. Las Vegas: IEEE Computer Society, 2016, pp. 1568-1574. ISBN 978-0-7695-4989-7.

[10]. zheng tang,gaoang Wang, Hao Xiao, Aotian Zheng and Jenq-Neng Hwang “single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features” top 1 NVIDIA AI CITY Challenge 2018

[11]. Ahmad Arinaldi et al “Detection and classification of vehicles for traffic video analytics” *Procedia Computer Science*, Volume 144, 2018

[12]. F. Zhang, C. Li, and F. Yang, “Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation,” *Sensors*, vol. 19, no. 3, p. 594, 2019.

[13] Vehicle speed tracking using OpenCV - Kenneth S. Palacio Baus <https://www.youtube.com/watch?v=66Pv5jnxUVI> (truy cập 12/2021).

[14]. He K., Zhang X., Ren S., et al. (2015). Deep Residual Learning for Image Recognition. *ArXiv151203385 Cs*.

[15]. A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, “Detection and classification of vehicles for traffic video analytics,” *Procedia Comput. Sci.*, vol. 144, pp. 259–268, 2018.

[16] Girshick, Ross, et al. "*Region-based convolutional networks for accurate object detection and segmentation.*" *IEEE transactions on pattern analysis and machine intelligence* 38.1 (2016): 142-158.

[17] Girshick, Ross. "*Fast r-cnn.*" *Proceedings of the IEEE international conference on computer vision*. 2015.

[18] Ren, Shaoqing, et al. "*Faster r-cnn: Towards real-time object detection with region proposal networks.*" *Advances in neural information processing systems*. 2015.

[19] Liu, Wei, et al. "*Ssd: Single shot multibox detector.*" *European conference on computer vision*. Springer, Cham, 2016.

[20] SPP-net] Dalal, Navneet, and Bill Triggs. "*Histograms of oriented gradients for human detection.*" *Computer Vision and Pattern Recognition*, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[21] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.

[22] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, IEEE, 2016.

[23] N. Wojke, A. Bewley, and D. Paulus, "*Simple online and realtime tracking with a deep association metric*," in *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649, IEEE, 2017.

[24] N. Wojke and A. Bewley, "Deep cosine metric learning for person reidentification," in *2018 IEEE winter conference on applications of computervision (WACV)*, pp. 748–756, IEEE, 2018.

[25] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European Conference on Computer Vision*, pp. 474–490, Springer, 2020.

[26] techzizou <https://github.com/techzizou>

[27] qfgaohao, "MobileNetV1, MobileNetV2, VGG based SSD/SSD-lite implementation," [Online]. Available: <https://github.com/qfgaohao/pytorch-ssd>.

## PHỤ LỤC

### 1. Source code train SSD trên Colab

```

from absl import flags

import tensorflow.compat.v2 as tf

from object_detection import model_lib_v2

flags.DEFINE_string('pipeline_config_path', None, 'Path to pipeline config "
file.>')

flags.DEFINE_integer('num_train_steps', None, 'Number of train steps.>')

flags.DEFINE_bool('eval_on_train_data', False, 'Enable evaluating on train '
                  'data (only supported in distributed training).>')

flags.DEFINE_integer('sample_1_of_n_eval_examples', None, 'Will sample
one of '
                  'every n eval input examples, where n is provided.>')

flags.DEFINE_integer('sample_1_of_n_eval_on_train_examples', 5, 'Will sa
mple '
                  'one of every n train input examples for evaluation, '
                  'where n is provided. This is only used if '
                  '`eval_training_data` is True.>')

flags.DEFINE_string(
    'model_dir', None, 'Path to output model directory '
    'where event and checkpoint files will be written.>')

flags.DEFINE_string(
    'checkpoint_dir', None, 'Path to directory holding a checkpoint. If '
    '`checkpoint_dir` is provided, this binary operates in eval-only mode, '
    'writing resulting metrics to `model_dir`.')

```



```
flags.DEFINE_integer('eval_timeout', 3600, 'Number of seconds to wait for a
n'
```

```
        'evaluation checkpoint before exiting.')
```

```
flags.DEFINE_bool('use_tpu', False, 'Whether the job is executing on a TPU.
')
```

```
flags.DEFINE_string(
```

```
    'tpu_name',
```

```
    default=None,
```

```
    help='Name of the Cloud TPU for Cluster Resolvers.')
```

```
flags.DEFINE_integer(
```

```
    'num_workers', 1, 'When num_workers > 1, training uses '
```

```
    'MultiWorkerMirroredStrategy. When num_workers = 1 it uses '
```

```
    'MirroredStrategy.')
```

```
flags.DEFINE_integer(
```

```
    'checkpoint_every_n', 1000, 'Integer defining how often we checkpoint.')
```

```
flags.DEFINE_boolean('record_summaries', True,
```

```
    ('Whether or not to record summaries defined by the model'
```

```
    ' or the training pipeline. This does not impact the'
```

```
    ' summaries of the loss values which are always'
```

```
    ' recorded.'))
```

```
FLAGS = flags.FLAGS
```

```
def main(UNUSED_argv):
```

```
    flags.mark_flag_as_required('model_dir')
```

```
    flags.mark_flag_as_required('pipeline_config_path')
```

```

tf.config.set_soft_device_placement(True)

if FLAGS.checkpoint_dir:
    model_lib_v2.eval_continuously(
        pipeline_config_path=FLAGS.pipeline_config_path,
        model_dir=FLAGS.model_dir,
        train_steps=FLAGS.num_train_steps,
        sample_1_of_n_eval_examples=FLAGS.sample_1_of_n_eval_examples,
s,
        sample_1_of_n_eval_on_train_examples=(
            FLAGS.sample_1_of_n_eval_on_train_examples),
        checkpoint_dir=FLAGS.checkpoint_dir,
        wait_interval=300, timeout=FLAGS.eval_timeout)
else:
    if FLAGS.use_tpu:
        # TPU is automatically inferred if tpu_name is None and
        # we are running under cloud ai-platform.
        resolver = tf.distribute.cluster_resolver.TPUClusterResolver(
            FLAGS.tpu_name)
        tf.config.experimental_connect_to_cluster(resolver)
        tf.tpu.experimental.initialize_tpu_system(resolver)
        strategy = tf.distribute.experimental.TPUStrategy(resolver)
    elif FLAGS.num_workers > 1:
        strategy = tf.distribute.experimental.MultiWorkerMirroredStrategy()
    else:
        strategy = tf.compat.v2.distribute.MirroredStrategy()

```

```

with strategy.scope():
    model_lib_v2.train_loop(
        pipeline_config_path=FLAGS.pipeline_config_path,
        model_dir=FLAGS.model_dir,
        train_steps=FLAGS.num_train_steps,
        use_tpu=FLAGS.use_tpu,
        checkpoint_every_n=FLAGS.checkpoint_every_n,
        record_summaries=FLAGS.record_summaries)

if __name__ == '__main__':
    tf.compat.v1.app.run()

```

**kết quả NUM\_TRAIN\_STEPS=10000**

```

INFO:tensorflow:Step 10000 per-step time 0.207s
I0527 10:28:59.226914 139929104537472 model_lib_v2.py:707] Step
10000 per-step time 0.207s
INFO:tensorflow: {'Loss/classification_loss': 0.046244226,
'Loss/localization_loss': 0.014821663,
'Loss/regularization_loss': 0.10360042,
'Loss/total_loss': 0.16466631,
'learning_rate': 0.07352352}
I0527 10:28:59.227187 139929104537472 model_lib_v2.py:708]
{'Loss/classification_loss': 0.046244226,
'Loss/localization_loss': 0.014821663,
'Loss/regularization_loss': 0.10360042,
'Loss/total_loss': 0.16466631,
'learning_rate': 0.07352352}

```

## 2. Source code Giao diện hiển thị

```

def main(sess=sesser):

    if True:
        fTime=time.time()
        _,image_np=cap.read(0)
        #image_np = imutils.resize(image_np, width=400)

        # Definite input and output Tensors for detection_graph

        # Expand dimensions since the model expects images to have shape: [1,
None, None, 3]
        image_np_expanded = np.expand_dims(image_np, axis=0)
        # Actual detection.
        (boxes, scores, classes, num) = sess.run(
            [detection_boxes, detection_scores, detection_classes, num_detections],
            feed_dict={image_tensor: image_np_expanded})

        # Visualization of the results of a detection.
        img=image_np
        imgF,coords=vis_util.visualize_boxes_and_labels_on_image_array(
            image_np,
            np.squeeze(boxes),
            np.squeeze(classes).astype(np.int32),
            np.squeeze(scores),
            category_index,
            use_normalized_coordinates=True,
            line_thickness=2)

        matchVehicles(coords,im_width,im_height,imgF)
        checkRedLightCrossed(imgF)
        checkSpeed(fTime,img)
        for v in vehicles:
            if v.getTracking()==True:

                for p in v.getPoints():
                    cv2.circle(image_np,p,3,(200,150,75),6)

```

```

#print(ymin*im_height,xmin*im_width,ymax*im_height,xmax*im_width)

#cv2.rectangle(image_np,(int(xmin*im_width),int(ymin*im_height)),(int(xmax
*im_width),int(ymax*im_height)),(255,0,0),2)
cv2.line(image_np, (int(xl1),int(yl1)), (int(xl2),int(yl2)), (0,255,0),3)
cv2.line(image_np, (int(xl3),int(yl3)), (int(xl4),int(yl4)), (0,0,255),3)
cv2.line(image_np, (int(xl5),int(yl5)), (int(xl6),int(yl6)), (255,0,0),3)
VideoFileOutput.write(image_np)
#print('yola')
frame=cv2.resize(image_np,(1020,647))
cv2image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGBA)
img = Image.fromarray(cv2image)
imgtk = ImageTk.PhotoImage(image=img)
display1.imgtk = imgtk #Shows frame for display 1
display1.configure(image=imgtk)
window.after(1, main)
lbl1 = tk.Label(frame,text='Khung hình chính để kết nối với camera',font =
"verdana 12 bold")
lbl1.pack(side='top')

lbl2 = tk.Label(frame2,text='Hình ảnh phương tiện vượt tốc độ',font = "verdana
10 bold")
lbl2.grid(row=0,column=0,sticky = 'S',pady=10)

lbl3 = tk.Label(frame3,text='Tốc độ ',font = "verdana 10 bold")
lbl3.grid(row=0,column=0,sticky = 'S',pady=10)

```

### 3. Source code xử lý video

```

detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.compat.v1.GraphDef()
    with tf.compat.v1.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
        tf.import_graph_def(od_graph_def, name="")
label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
categories = label_map_util.convert_label_map_to_categories(label_map,
max_num_classes=NUM_CLASSES, use_display_name=True)
category_index = label_map_util.create_category_index(categories)

```

```

cap=cv2.VideoCapture('ssssabc1.mp4') # 0 stands for very first webcam attach
#cap=cv2.VideoCapture('abc.mp4')
filename="testoutput.avi"
codec=cv2.VideoWriter_fourcc('m','p','4','v')#fourcc stands for four character code
framerate=10
resolution=(640,480)
VideoFileOutput=cv2.VideoWriter(filename,codec,framerate, resolution)
vs = WebcamVideoStream(src='Set01_video01.mp4').start()
ret,imgF=cap.read(0)
imgF=Image.fromarray(imgF)
im_width, im_height = imgF.size
x11=0
x12=im_width-1
y11=im_height*0.5
y12=y11
m11=(y12-y11)/(x12-x11)
intcpt1=y11-m11*x11

count=0
x13=0
x14=im_width-1
y13=im_height*0.25
y14=y13
m12=(y14-y13)/(x14-x13)
intcpt2=y13-m12*x13

x15=0
x16=im_width-1
y15=im_height*0.1
y16=y15
m13=(y16-y15)/(x16-x15)
intcpt3=y15-m13*x15
ret=True
start=time.time()
c=0
sesser=tf.compat.v1.Session(graph=detection_graph)
image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
detection_boxes = detection_graph.get_tensor_by_name('detection_boxes:0')
detection_scores = detection_graph.get_tensor_by_name('detection_scores:0')
detection_classes = detection_graph.get_tensor_by_name('detection_classes:0')

```

```

num_detections = detection_graph.get_tensor_by_name('num_detections:0')
window = tk.Tk() #Makes main window
window.wm_title("T.M.S")
window.columnconfigure(0, {'minsize': 1020})
window.columnconfigure(1, {'minsize': 335})
frame=tk.Frame(window)
frame.grid(row=0,column=0,rowspan=5,sticky='N',pady=10)
frame2=tk.Frame(window)
frame2.grid(row=0,column=1)
frame3=tk.Frame(window)
frame3.grid(row=1,column=1)
frame4=tk.Frame(window)
frame4.grid(row=2,column=1)
frame5=tk.Frame(window)
frame5.grid(row=3,column=1)
frame2.rowconfigure(1, {'minsize': 250})
frame3.rowconfigure(1, {'minsize': 80})
frame4.rowconfigure(1, {'minsize': 150})
frame5.rowconfigure(1, {'minsize': 80})

```

#### 4. Source code tính tốc độ của phương tiện và lưu hình ảnh phương tiện vượt giới hạn

```

def checkSpeed(ftime,img):
    for v in vehicles:
        if v.speedChecked==False and len(v.points)>=2:
            x1,y1=v.points[0]
            x2,y2=v.points[-1]
            if y2<y11 and y2>y13 and v.entered==False:
                v.enterTime=ftime
                v.entered=True
            elif y2<y13 and y2 > y15 and v.exited==False:
                v.exitTime=ftime
                v.exited==False
                v.speedChecked=True
                speed=25/(v.exitTime-v.enterTime)
                print(speed)
                bimg=img[int(v.rect[1]):int(v.rect[1]+v.rect[3]),
int(v.rect[0]):int(v.rect[0]+v.rect[2])]
                frame2=bimg
                img2 = Image.fromarray(frame2)

```

```

w,h=img2.size
asprto=w/h
frame2=cv2.resize(frame2,(250,int(250/asprto)))
cv2image2 = cv2.cvtColor(frame2, cv2.COLOR_BGR2RGBA)
img2 = Image.fromarray(cv2image2)
imgtk2 = ImageTk.PhotoImage(image=img2)
display2.imgtk = imgtk2 #Shows frame for display 1
display2.configure(image=imgtk2)
display3.configure(text=str(speed)[:5]+'Km/hr')
if speed >20: -----# test
    speedcv=round(speed,1)
    #cv2.imshow('BROKE',bimg)
    #name='Rule Breakers/culprit'+ '-' +str(speed)+'-
'+str(time.time())+'.jpg'
    name = 'Rule Breakers/culprit' + '-' + str(speed) + '-' +
str(time.time()) + '.jpg'
    cv2.imwrite(name,bimg)
    tstop = threading.Event()
    thread = threading.Thread(target=getLicensePlateNumber,
args=(name,))
    thread.daemon = True
    thread.start()
detection_graph = tf.Graph()
with detection_graph.as_default():
    od_graph_def = tf.compat.v1.GraphDef()
    with tf.compat.v1.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
        serialized_graph = fid.read()
        od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name=")

```